

BUYPASS CODE MANAGER

REST-API

Buypass OPEN

Version: 1.1
Document date: 26.03.2014

Buypass AS

Nydalsveien 30A, PO Box 4364 Nydalen
N-0402 Oslo, Norway

Tel.: +47 23 14 59 00
Fax: +47 23 14 59 01

E-mail: kundeservice@buypass.no
VAT: NO 983 163 327

www.buypass.no

History of change

Version	Date	Status	Description/Change
1.0	14.06.2012	First version	Created
1.1	25.03.2014	Updated version	Updated with new features
1.2	15.09.2014	Updated version	Updated with new HTTP status codes and error messages

Contributors

Company name	Name
Buypass AS	Jan-Olav Eide
Buypass AS	Oskar Otterskog

Table of content

Description of REST-API.....	5
1 Authentication	6
1.1 Phase 1	6
1.2 Phase 2	6
2 Event log	7
2.1 GET event-log	7
3 Audit log	8
3.1 GET audit-log	8
4 Profile	10
4.1 GET profile	10
4.2 GET profile/{username}.....	12
4.3 DELETE profile/{username}.....	13
4.4 POST profile/{username}/activate.....	13
4.5 POST profile/{username}/clearcache.....	14
4.6 POST profile/{username}/resetusertype	14
4.7 POST profile/{username}/offline	14
5 Unblock	14
5.1 GET unblock	14
5.2 GET unblock/{appld:telNr}/ubc	15
5.3 GET unblock/{appld:telNr}/regenerateUbc	16
6 Merchant.....	17
6.1 GET merchant.....	17
6.2 PUT merchant	19
7 Admin	19
7.1 GET admin	19
7.2 GET admin/{username}.....	20
7.3 POST admin.....	21
7.4 PUT admin	21
7.5 DELETE admin/{username}.....	21
7.6 GET admin/{username}/deactivate	21
7.7 GET admin/{username}/activate	21
8 LDAP.....	21
8.1 GET ldap	21
8.2 GET ldap/{id}.....	23
8.3 POST ldap.....	24
8.4 PUT ldap/{id}	24
8.5 DELETE ldap/{id}	24
8.6 GET ldap/{id}/up.....	24
8.7 GET ldap/{id}/down	24
8.8 GET ldap/{id}/ldap-path.....	25
8.9 GET ldap/{id}/ldap-path/{priority}	25
8.10 POST ldap/{id}/ldap-path.....	26
8.11 PUT ldap/{id}/ldap-path/{priority}	26
8.12 DELETE ldap/{id}/ldap-path/{priority}	26
8.13 GET cache.....	26
9 Radius.....	26
9.1 GET radius	26
9.2 GET radius/{id}.....	29
9.3 POST radius.....	31
9.4 PUT radius/{id}	31
9.5 DELETE radius/{id}	31
10 External Radius.....	32
10.1 GET external-radius	32
10.2 GET external-radius/{id}	32
10.3 POST external-radius	33

10.4 PUT external-radius/{id} 33
10.5 DELETE external-radius/{id} 34

Description of REST-API

Buypass Code REST-API allows merchants to integrate with Buypass Code Manager programmatically via HTTPS-requests. This is the official documentation of that functionality.

Requests and responses from the server can be in either XML- or JSON. The merchant can decide which format is best to use.

For PUT/POST-requests the same object structure is used as the one received from GET-requests, if nothing else is specified. For compatibility with systems that doesn't support PUT/DELETE-requests the API accepts a parameter “_method” which specifies the HTTP-method to use, e.g.

```
https://www.buypass.no/bpcode/merchants/{merchant-id}/ldap/{id}.{json,  
xml}?_method=DELETE
```

Successful requests will return an HTTP status code of 200.

1 Error handling

The API uses HTTP status codes and error messages to inform the user about errors that occur.

1.1 Validation errors

If validation of a field fails the HTTP status code will be 400.

Response

JSON example:

```
{  
  "actionErrors": [],  
  "fieldErrors": {  
    "appId": [  
      "You have to enter a numerical APP ID."  
    ]  
  }  
}
```

XML example:

```
1. <map>  
2.   <entry>  
3.     <string>actionErrors</string>  
4.     <list/>  
5.   </entry>  
6.   <entry>  
7.     <string>fieldErrors</string>  
8.     <linked-hash-map>  
9.       <entry>  
10.        <string>appId</string>  
11.        <list>  
12.          <string>You have to enter a numerical APP ID.</string>  
13.        </list>  
14.      </entry>  
15.    </linked-hash-map>  
16.  </entry>  
17. </map>
```

1.2 Invalid syntax

If XML or JSON syntax is invalid HTTP status code will be 400.

Response

JSON example:

```
{
  "errors": {
    "message": [
      "Malformed JSON"
    ]
  }
}
```

XML example:

```
1. <map>
2.   <entry>
3.     <string>errors</string>
4.     <linked-hash-map>
5.       <entry>
6.         <string>message</string>
7.         <list>
8.           <string>Malformed XML</string>
9.         </list>
10.      </entry>
11.    </linked-hash-map>
12.  </entry>
13. </map>
```

2 Authentication

Authentication is performed using an enterprise certificate which is issued by Buypass. The authentication is divided into two phases, explained below.

2.1 Phase 1

In phase 1 a GET-request is made to get a *logonChallenge*.

URL: `https://www.buypass.no/bpcode/merchants/{merchant-id}/logon.{json, xml}`

Response

JSON example:

```
{
  "logonMethod": "",
  "certificate": "",
  "challenge": "yh4uq+8jE2RU2jB2DMnXhfz/110=",
  "signature": ""
}
```

XML example:

```
1. <Logon>
2.   <challenge>xKjpOb2MOX01+NGU7oVhUNXuQcs=</challenge>
3. </Logon>
```

2.2 Phase 2

In phase 2 a POST-request is made to the same URL as in phase 1, and has to contain a signed message and the certificate that was used for signing. The message that is signed is an assembly of the merchant's ID, the challenge from phase 1, and the string "login". The fields are separated by "|", e.g. *Signature = B64encode(certificate.sign(merchantId|yh4uq+8jE2RU2jB2DMnXhfz/110|login))*

JSON example phase 2 request:

```
{
  "logonMethod": "pki",
```

```
"certificate": "B64-encoded certificate",
"signature": "B64-encoded signature"
}
```

XML example phase 2 request:

```
1. <Logon>
2.   <loginMethod>pki</loginMethod>
3.   <certificate>B64-encoded certificate</certificate>
4.   <signature>B64-encoded signature</signature>
5. </Logon>
```

Message fields

Fields	Description
loginMethod	Has to be "pki" in a phase 2 request. Can be ignored in phase 1.
certificate	B64-encoded certificate in phase 2 request. Can be ignored in phase 1.
challenge	Challenge. Returned in phase 1 response. Should not be included in phase 2 request.
signature	B64-encoded signature in phase 2 request. Can be ignored in phase 1.

If phase 2 is successful a session cookie will be returned. Clients must therefore allow session cookies to be set and included in later requests.

3 Event log

3.1 GET event-log

Get information about logins, attempts and errors.

URL: [https://www.bypass.no/bpcode/merchants/{merchant-id}/event-log.{json,xml}\[?arg1=val1\]](https://www.bypass.no/bpcode/merchants/{merchant-id}/event-log.{json,xml}[?arg1=val1])

Parameters

Name	Type	Description
fromDate	long	From date in UNIX timestampformat
toDate	long	To date in UNIX timestampformat
pagingFrom	int	Pagination, first row number
pagingTo	int	Pagination, last row number
user	string	Username
desc	string	Description. Can also be part of description
type	string	ACCESS_DENIED, ACCESS_GRANTED,INFO,ERROR, EXTERNAL_ACCESS_GRANTED, EXTERNAL_ACCESS_DENIED, EXTERNAL_INFO, EXTERNAL_ERROR, ACCESS_GRANTED_OFFLINE, ACCESS_DENIED_OFFLINE or ALL

Response

JSON example:

```
[
  {
```

```

    "description": "Administrator authenticated",
    "eventType": "ACCESS_GRANTED",
    "logDate": {
      "date": 26,
      "day": 3,
      "hours": 8,
      "minutes": 35,
      "month": 2,
      "seconds": 28,
      "time": 1395819328000,
      "timezoneOffset": -60,
      "year": 114
    },
    "logEntryId": 433986,
    "merchantId": 666667,
    "username": "oott"
  }
]

```

XML example:

```

1. <list>
2. <EventLogEntry>
3. <logEntryId class="long">433986</logEntryId>
4. <merchantId>666667</merchantId>
5. <logDate>2014-03-26 07:35:28.0 UTC</logDate>
6. <username>oott</username>
7. <description>Administrator authenticated</description>
8. <eventType>ACCESS_GRANTED</eventType>
9. </EventLogEntry>
10. </list>

```

Response fields

Fields	Description
logEntryId	Log object id
merchantId	Merchant's id
logDate	Date when log object was created
username	Username of the user that caused the event
eventType	One of ACCESS_DENIED, ACCESS_GRANTED, INFO, ERROR, EXTERNAL_ACCESS_GRANTED, EXTERNAL_ACCESS_DENIED, EXTERNAL_INFO, EXTERNAL_ERROR, ACCESS_GRANTED_OFFLINE or ACCESS_DENIED_OFFLINE
description	Description of the event

4 Audit log

4.1 GET audit-log

Get information about changes performed by admins.

URL: [https://www.bypass.no/bpcode/merchants/{merchant-id}/audit-log.{json,xml}\[?arg1=val1\]](https://www.bypass.no/bpcode/merchants/{merchant-id}/audit-log.{json,xml}[?arg1=val1])

Parameters

Name	Type	Description
------	------	-------------

fromDate	long	From date in UNIX timestampformat
toDate	long	To date in UNIX timestampformat
pagingFrom	int	Pagination, first row number
pagingTo	int	Pagination, last row number
user	string	Username of admin
desc	string	Description. Can also be part of description
type	string	RADIUS_CREATE, RADIUS_EDIT, RADIUS_DELETE, LDAP_CREATE, LDAP_EDIT, LDAP_DELETE, LDAP_PATH_CREATE, LDAP_PATH_EDIT, LDAP_PATH_DELETE, ADMIN_CREATE, ADMIN_EDIT, ADMIN_DELETE, USER_STATUS, UNBLOCK_GENERATED, LDAP_PRIORITY_EDIT, LDAP_CACHE_CLEAR, LDAP_USER_CACHE_CLEAR, RESET_USER_TYPE, EXTERNAL_RADIUS_CREATE, EXTERNAL_RADIUS_EDIT, EXTERNAL_RADIUS_DELETE or OFFLINE_EDIT

Response

JSON example:

```
[
  {
    "description": "Reset user type for oott",
    "eventType": "RESET_USER_TYPE",
    "logDate": {
      "date": 25,
      "day": 2,
      "hours": 15,
      "minutes": 51,
      "month": 2,
      "seconds": 58,
      "time": 1395759118000,
      "timezoneOffset": -60,
      "year": 114
    },
    "logEntryId": 4605,
    "merchantId": 666667,
    "username": "oott"
  }
]
```

XML example:

```
1. <list>
2. <AuditLogEntry>
3. <logEntryId class="long">4605</logEntryId>
4. <merchantId>666667</merchantId>
5. <logDate>2014-03-25 14:51:58.0 UTC</logDate>
6. <username>oott</username>
7. <description>Reset user type for oott</description>
8. <eventType>RESET_USER_TYPE</eventType>
9. </AuditLogEntry>
10. </list>
```

Response fields

Fields	Description
--------	-------------

logEntryId	Log object id
merchantId	Merchant's id
logDate	Date when log object was created
username	Username of the user that caused the event
eventType	RADIUS_CREATE, RADIUS_EDIT, RADIUS_DELETE, LDAP_CREATE, LDAP_EDIT, LDAP_DELETE, LDAP_PATH_CREATE, LDAP_PATH_EDIT, LDAP_PATH_DELETE, ADMIN_CREATE, ADMIN_EDIT, ADMIN_DELETE, USER_STATUS, UNBLOCK_GENERATED, LDAP_PRIORITY_EDIT, LDAP_CACHE_CLEAR, LDAP_USER_CACHE_CLEAR, RESET_USER_TYPE, EXTERNAL_RADIUS_CREATE, EXTERNAL_RADIUS_EDIT, EXTERNAL_RADIUS_DELETE or OFFLINE_EDIT
description	Description of the event

5 Profile

5.1 GET profile

Get the users that have used Bypass Code to login. The information has been fetched from an LDAP directory.

URL: `https://www.bypass.no/bpcode/merchants/{merchant-id}/profile.{json,xml}[?arg1=val1]`

Parameters

Name	Type	Description
filter	string	Filter using status. One of active, inactive or all
search	string	Search by username. Either whole username or part of username
pagingFrom	int	Pagination, first row number
pagingTo	int	Pagination, last row number

Response

JSON example:

```
[
  {
    "cachedAppId": 0,
    "cachedPhoneMobile": "12345678",
    "created": {
      "date": 6,
      "day": 4,
      "hours": 12,
      "minutes": 57,
      "month": 2,
      "seconds": 28,
      "time": 1394107048000,
      "timezoneOffset": -60,
      "year": 114
    },
    "fqdn": "cn=oott,dc=example,dc=com",
    "lastFailed": {
      "date": 24,
      "day": 1,

```

```

        "hours":16,
        "minutes":39,
        "month":2,
        "seconds":56,
        "time":1395675596000,
        "timezoneOffset":-60,
        "year":114
    },
    "lastFailedExternal":null,
    "lastSuccessful":{
        "date":10,
        "day":1,
        "hours":8,
        "minutes":27,
        "month":2,
        "seconds":8,
        "time":1394436428000,
        "timezoneOffset":-60,
        "year":114
    },
    "lastSuccessfulExternal":null,
    "lifecycle":20,
    "merchantId":666666,
    "modified":null,
    "uname":"oott",
    "userType":1
}
]

```

XML example:

```

1. <list>
2. <User>
3. <uname>oott</uname>
4. <lifecycle class="long">20</lifecycle>
5. <merchantId class="long">666666</merchantId>
6. <fqdn>cn=oott,dc=example,dc=com</fqdn>
7. <created>2014-03-06 11:57:28.0 UTC</created>
8. <userType class="int">1</userType>
9. <lastSuccessful>2014-03-10 07:27:08.0 UTC</lastSuccessful>
10. <lastFailed>2014-03-24 15:39:56.0 UTC</lastFailed>
11. <cachedPhoneMobile>12345678</cachedPhoneMobile>
12. </User>
13. </list>

```

Response fields

Fields	Description
uname	Username of the user
lifecycle	Status of user. Active (20) or inactive (83)
merchantId	Merchant's id
fqdn	Fully qualified name of where user was last found in LDAP directory
created	Date when user first logged in
cachedPhoneMobile	User's phone number that was last found in LDAP directory
cachedAppld	User's appld that was last found in LDAP directory
modified	Date when status of user was last changed
lastSuccessful	Date of last successful BP Code authentication

lastFailed	Date of last failed BP Code authentication
lastSuccessfulExternal	Date of last successful authentication using external radius
lastFailedExternal	Date of last failed authentication using external radius
userType	0=No successful authentication, 1=Has at least one successful BP Code authentication, 2=Has at least one successful external radius authentication but no successful BP Code authentication

5.2 GET profile/{username} ¹

Get a user that has used Bypass Code to login. The information has been fetched from an LDAP directory.

URL: `https://www.bypass.no/bpcode/merchants/{merchant-id}/profile/{username}.{json, xml}[?arg1=val1]`

Response

JSON example:

```
{
  "cachedAppId":0,
  "cachedPhoneMobile":"12345678",
  "created":{
    "date":6,
    "day":4,
    "hours":12,
    "minutes":57,
    "month":2,
    "seconds":28,
    "time":1394107048000,
    "timezoneOffset":-60,
    "year":114
  },
  "fqdn":"cn=oott,dc=example,dc=com",
  "lastFailed":{
    "date":24,
    "day":1,
    "hours":16,
    "minutes":39,
    "month":2,
    "seconds":56,
    "time":1395675596000,
    "timezoneOffset":-60,
    "year":114
  },
  "lastFailedExternal":null,
  "lastSuccessful":{
    "date":10,
    "day":1,
    "hours":8,
    "minutes":27,
    "month":2,
    "seconds":8,
    "time":1394436428000,
    "timezoneOffset":-60,
    "year":114
  },
}
```

¹ Available from release 7.27.X.X, June 2014

```

    "lastSuccessfulExternal":null,
    "lifecycle":20,
    "merchantId":666666,
    "modified":null,
    "uname":"oott",
    "userType":1
  }

```

XML example:

```

1. <User>
2.   <uname>oott</uname>
3.   <lifecycle class="long">20</lifecycle>
4.   <merchantId class="long">666666</merchantId>
5.   <fqdn>cn=oott,dc=example,dc=com</fqdn>
6.   <created>2014-03-06 11:57:28.0 UTC</created>
7.   <userType class="int">1</userType>
8.   <lastSuccessful>2014-03-10 07:27:08.0 UTC</lastSuccessful>
9.   <lastFailed>2014-03-24 15:39:56.0 UTC</lastFailed>
10.  <cachedPhoneMobile>12345678</cachedPhoneMobile>
11. </User>

```

Response fields

Fields	Description
uname	Username of the user
lifecycle	Status of user. Active (20) or inactive (83)
merchantId	Merchant's id
fqdn	Fully qualified name of where user was last found in LDAP directory
created	Date when user first logged in
cachedPhoneMobile	User's phone number that was last found in LDAP directory
cachedAppld	User's appld that was last found in LDAP directory
modified	Date when status of user was last changed
lastSuccessful	Date of last successful BP Code authentication
lastFailed	Date of last failed BP Code authentication
lastSuccessfulExternal	Date of last successful authentication using external radius
lastFailedExternal	Date of last failed authentication using external radius
userType	0=No successful authentication, 1=Has at least one successful BP Code authentication, 2=Has at least one successful external radius authentication but no successful BP Code authentication

5.3 DELETE profile/{username}

Deactivate a user

URL: <https://www.bypass.no/bpcode/merchants/{merchant-id}/profile/{username}.{json, xml}>

5.4 POST profile/{username}/activate

Activate a user

URL: <https://www.bypass.no/bpcode/merchants/{merchant-id}/profile/{username}/activate.{json, xml}>

5.5 POST profile/{username}/clearcache

Empty ldap cache for a user

URL: `https://www.buypass.no/bpcode/merchants/{merchant-id}/profile/{username}/clearcache.{json, xml}`

5.6 POST profile/{username}/resetusertype

Reset user type for a user

URL: `https://www.buypass.no/bpcode/merchants/{merchant-id}/profile/{username}/resetusertype.{json, xml}`

5.7 POST profile/{username}/offline

Sync user client time offset

URL: `https://www.buypass.no/bpcode/merchants/{merchant-id}/profile/{username}/offline.{json, xml}`

Parameters

Name	Type	Description
clientUnixTime	int	Unixtime in seconds given by users application
clientAppld	int	User's client's appld

6 Unblock

6.1 GET unblock

Get unblock status of a user or an administrator.

URL: `https://www.buypass.no/bpcode/merchants/{merchant-id}/unblock.{json, xml}[?arg1=val1]`

Parameters

Name	Type	Description
profiles	string	Specifies which users' unblock status to get. Specified in the format "appld:phoneNumber". It is possible to get several users' unblock status by separating them using ",". Only one of appld and phoneNumber has to be used. E.g. "+4712457898"

Response

JSON example:

```
[
  {
    "appId": "1000068393",
    "merchantId": "",
    "phoneNumber": "42398745",
    "unblockCode": "",
    "unblockStatus": "20"
  }
]
```

XML example:

1. [<list>](#)

```

2. <UnblockStatus>
3. <appId>1000068393</appId>
4. <phoneNumber>42398745</phoneNumber>
5. <unlockStatus>20</unlockStatus>
6. </UnblockStatus>
7. </list>
    
```

Response fields

Fields	Description
appId	User's appId
phoneNumber	User's phone number
unlockStatus	User's unlock status: active (10), user blocked (20), unlock code used (40) and unlock code is blocked (80)
unlockCode	Not used in this response
merchantId	Not used in this response

6.2 GET unblock/{appId:telNr}/ubc

Get unlock code for a user or an administrator that is blocked.

URL: `https://www.bypass.no/bpcode/merchants/{merchant-id}/unlock/{appId:phoneNumber}/ubc.{json, xml}`

Response

JSON example:

```

{
  "appId": "1000068393",
  "merchantId": "",
  "phoneNumber": "42398745",
  "unlockCode": "035065004",
  "unlockStatus": ""
}
    
```

XML example:

```

1. <UnblockStatus>
2. <appId>1000068393</appId>
3. <phoneNumber>42398745</phoneNumber>
4. <unlockCode>035065004</unlockCode>
5. </UnblockStatus>
    
```

Response fields

Fields	Description
appId	User's appId
phoneNumber	User's phone number
unlockStatus	Not used in this response
unlockCode	User's unlock code
merchantId	Not used in this response

6.3 GET unblock/{appld:telNr}/regenerateUbc

Generate a new unblock code for a user or an administrator who has blocked his/her unblock code.

URL: `https://www.bypass.no/bpcode/merchants/{merchant-id}/unblock/{appId:telNr}/regenerateUbc.{json, xml}`

7 Offline Unlock

7.1 GET offline-unlock

Get offline unlock status of a user.

URL: `https://www.bypass.no/bpcode/merchants/{merchant-id}/offline-unlock.{json, xml}[?arg1=val1]`

Parameters

Name	Type	Description
profiles	string	Specifies which users' offline unlock status to get. Specified in the format "appld:phoneNumber". It is possible to get several users' offline unlock status by separating them using ",". Only one of appld and phoneNumber has to be used. E.g. "+4712457898"

Response

JSON example:

```
[
  {
    "appId": "1000068393",
    "merchantId": "",
    "phoneNumber": "42398745",
    "unblockCode": "",
    "unblockStatus": "20"
  }
]
```

XML example:

```
8. <list>
9.   <UnblockStatus>
10.    <appId>1000068393</appId>
11.    <phoneNumber>42398745</phoneNumber>
12.    <unblockStatus>20</unblockStatus>
13.  </UnblockStatus>
14. </list>
```

Response fields

Fields	Description
appId	User's appld
phoneNumber	User's phone number
unblockStatus	User's unblock status: active (10), user blocked (20), unblock code used (40) and unblock code is blocked (80)
unblockCode	Not used in this response
merchantId	Not used in this response

7.2 GET unblock/{appId:telNr}/ubc

Get unblock code for a user or an administrator that is blocked.

URL: `https://www.bypass.no/bpcode/merchants/{merchant-id}/unblock/{appId:phoneNumber}/ubc.{json, xml}`

Response

JSON example:

```
{
  "appId": "1000068393",
  "merchantId": "",
  "phoneNumber": "42398745",
  "unlockCode": "035065004",
  "unlockStatus": ""
}
```

XML example:

```
6. <UnlockStatus>
7. <appId>1000068393</appId>
8. <phoneNumber>42398745</phoneNumber>
9. <unlockCode>035065004</unlockCode>
10. </UnlockStatus>
```

Response fields

Fields	Description
appId	User's appId
phoneNumber	User's phone number
unlockStatus	Not used in this response
unlockCode	User's unlock code
merchantId	Not used in this response

8 Merchant

8.1 GET merchant

Get merchant configuration and information. This information is used to show contact information when a user is blocked, which app types are allowed for the merchant, and the merchant's Service Connector(s).

URL: `https://www.bypass.no/bpcode/merchants/{merchant-id}/merchant.{json, xml}`

Response

JSON example:

```
{
  "addableAdminUsers": null,
  "adminUsers": null,
  "appConfigs": [
    {
      "appType": "100004",
      "merchantId": "666666"
    },
    {

```

```

    "appType": "120001",
    "merchantId": 666666
  },
  ],
  "defaultCountryCode": "+47",
  "description": "Oskar testbrukersted",
  "gateways": [
    {
      "name": "oott gateway",
      "version": "7.25.1.0"
    }
  ],
  ],
  "merchantContactInfo": "Ring oss! 97837085",
  "merchantId": 666667,
  "merchantName": "Demobrukersted"
}

```

XML example:

```

1. <Merchant>
2.   <merchantId class="big-decimal">666667</merchantId>
3.   <merchantName>Demobrukersted</merchantName>
4.   <merchantContactInfo>Ring oss! 97837085</merchantContactInfo>
5.   <description>Oskar testbrukersted</description>
6.   <defaultCountryCode>+47</defaultCountryCode>
7.   <appConfigs class="list">
8.     <AppConfig>
9.       <merchantId class="big-decimal">666666</merchantId>
10.      <appType>100004</appType>
11.     </AppConfig>
12.     <AppConfig>
13.       <merchantId class="big-decimal">666666</merchantId>
14.       <appType>120001</appType>
15.     </AppConfig>
16.   </appConfigs>
17.   <gateways class="list">
18.     <Gateway>
19.       <name>oott gateway</name>
20.       <version>7.25.1.0</version>
21.     </Gateway>
22.   </gateways>
23. </Merchant>

```

Response fields

Fields	Description
merchantId	Merchant's id
merchantName	Name of the merchant
merchantContactInfo	Merchant's contact information
description	Buypass' description of the merchant
appConfigs	An array of allowed app configurations for the merchant. An app configuration has a merchantId that belongs to the merchant that owns the app configuration; this could be either this merchant or this merchant's parent. An app configuration also has an appType field.
gateways	An array of the merchant's Service Connector(s). A gateway has a name field and a version field.
countryCode	Merchant's default country code
addableAdminUsers	Not in use

adminUsers	Not in use
------------	------------

8.2 PUT merchant

Update merchant information.

URL: `https://www.bypass.no/bpcode/merchants/{merchant-id}/merchant/{merchant-id}.{json, xml}`

Message fields

Fields	Description
merchantName	Name of the merchant
merchantContactInfo	Merchant's contact information

9 Admin

9.1 GET admin

Get all admins that have access to Bypass Code Manager.

URL: `https://www.bypass.no/bpcode/merchants/{merchant-id}/admin.{json, xml}`

Response

JSON example:

```
[
  {
    "appId": "1000068377",
    "lifeCycle": 20,
    "merchantId": 666666,
    "phoneNumber": "+4797837085",
    "role": "SUPERUSER",
    "uname": "oott",
    "unameEncoded": "oott",
    "userId": 1
  }
]
```

XML example:

```
1. <list>
2.   <AdminUser>
3.     <uname>oott</uname>
4.     <appId>1000068377</appId>
5.     <phoneMobile>+4797837085</phoneMobile>
6.     <lifecycle class="big-decimal">20</lifecycle>
7.     <merchantId class="big-decimal">666666</merchantId>
8.     <merchantAdminUserId class="int">1</merchantAdminUserId>
9.     <role>SUPERUSER</role>
10.    <firstName>John</firstName>
11.    <lastName>Doe</lastName>
12.    <email>doe@bypass.no</email>
13.    <other>Jobber for Bypass</other>
14.  </AdminUser>
15. </list>
```

Response fields

Fields	Description
--------	-------------

uname	Administrator's username
unameEncoded	Administrator's username with "." swapped with "DOT" and "@" swapped with "AT"
appld	Administrator's appld
phoneNumber	Administrator's phone number
lifeCycle	Status of the administrator: active (20), inactive (83)
merchantId	Merchant's Id
userId	Id of the administrator
role	Administrator's role: HELPDESK, USERADMIN or SUPERUSER
firstName	Administrator's first name
lastName	Administrator's last name
email	Administrator's e-mail
other	Other information

9.2 GET admin/{username}

Get an administrator with access to Bypass Code Manager.

URL: `https://www.bypass.no/bpcode/merchants/{merchant-id}/admin/{username}.{json, xml}`

Response

JSON example:

```
{
  "appId": "1000068377",
  "lifeCycle": 20,
  "merchantId": 666666,
  "phoneNumber": "+4797837085",
  "role": "SUPERUSER",
  "uname": "oott",
  "unameEncoded": "oott",
  "userId": 1
}
```

XML example:

```
1. <AdminUser>
2. <uname>oott</uname>
3. <appId>1000068377</appId>
4. <phoneMobile>+4797837085</phoneMobile>
5. <lifecycle class="big-decimal">20</lifecycle>
6. <merchantId class="big-decimal">666666</merchantId>
7. <merchantAdminUserId class="int">1</merchantAdminUserId>
8. <role>SUPERUSER</role>
9. <firstName>John</firstName>
10. <lastName>Doe</lastName>
11. <email>doe@bypass.no</email>
12. <other>Jobber for Bypass</other>
13. </AdminUser>
```

Response fields

Fields	Description
uname	Administrator's username

unameEncoded	Administrator's username with "." swapped with "DOT" and "@" swapped with "AT"
appld	Administrator's appld
phoneNumber	Administrator's phone number
lifeCycle	Status of the administrator: active (20), inactive (83)
merchantId	Merchant's Id
userId	Id of the administrator
role	Administrator's role: HELPDESK, USERADMIN or SUPERUSER
firstName	Administrator's first name
lastName	Administrator's last name
email	Administrator's e-mail
other	Other information

9.3 POST admin

Create a new administrator in Bypass Code Manager.

URL: <https://www.bypass.no/bpcode/merchants/{merchant-id}/admin.{json, xml}>

9.4 PUT admin

Update an existing administrator in Bypass Code Manager.

URL: <https://www.bypass.no/bpcode/merchants/{merchant-id}/admin.{json, xml}>

9.5 DELETE admin/{username}

Delete an administrator in Bypass Code Manager.

URL: <https://www.bypass.no/bpcode/merchants/{merchant-id}/admin/{username}.{json, xml}>

9.6 GET admin/{username}/deactivate

Deactivate an administrator in Bypass Code Manager.

URL: <https://www.bypass.no/bpcode/merchants/{merchant-id}/admin/{username}/deactivate.{json, xml}>

9.7 GET admin/{username}/activate

Activate an administrator in Bypass Code Manager.

URL: <https://www.bypass.no/bpcode/merchants/{merchant-id}/admin/{username}/activate.{json, xml}>

10 LDAP

10.1 GET ldap²

Get all LDAP configurations

URL: <https://www.bypass.no/bpcode/merchants/{merchant-id}/ldap.{json, xml}>

Response

² Available from release 7.27.X.X, June 2014

JSON example:

```
[
  {
    "appIdAttrName": "",
    "description": "LDAPBPLAB01",
    "enabled": true,
    "ldapConfigId": 4,
    "ldapPassword": "xxxxx",
    "ldapPaths": [
      {
        "ldapId": 4,
        "ldapPathId": 8,
        "priority": 1,
        "searchString": "basedn[ou=Users,ou=BPCode,dc=BPLAB01,dc=local(*)],filter[samAccountName=#USER#]"
      }
    ],
    "ldapUrl": "ldap://192.168.171.15",
    "ldapUsername": "bpc_svc_ldap",
    "lockoutDuration": 0,
    "merchantId": 0,
    "phoneNumberAttrName": "phoneNumber;mobile",
    "priority": 2,
    "securityRequirement": ""
  }
]
```

XML example:

```
1. <list>
2.   <LdapModel>
3.     <ldapConfigId class="int">4</ldapConfigId>
4.     <ldapUrl>ldap://192.168.171.15</ldapUrl>
5.     <ldapUsername>bpc_svc_ldap</ldapUsername>
6.     <ldapPassword>xxxxx</ldapPassword>
7.     <phoneNumberAttrName>phoneNumber;mobile</phoneNumberAttrName>
8.     <priority class="big-decimal">2</priority>
9.     <description>LDAP BPLAB01</description>
10.    <enabled>true</enabled>
11.    <lockoutDuration>0</lockoutDuration>
12.    <ldapPaths class="list">
13.      <LdapPathModel>
14.        <ldapConfigId class="int">4</ldapConfigId>
15.        <ldapPathId class="int">8</ldapPathId>
16.        <priority class="big-decimal">1</priority>
17.        <searchString>basedn[ou=Users,ou=BPCode,dc=BPLAB01,dc=local(*)],filter[samAccountName=#USER#]</searchString>
18.      </LdapPathModel>
19.    </ldapPaths>
20.  </LdapModel>
21. </list>
```

Response fields

Fields	Description
ldapConfigId	LDAP configuration id
ldapUrl	URI to the LDAP directory
ldapPaths	Array of LDAP paths, see ldap-path section
priority	Priority of the LDAP configuration
merchantId	Deprecated
appIdAttrName	Attribute in LDAP directory that contains the user's appId

phoneNumberAttrName	Attribute in LDAP directory that contains the user's phone number
ldapUsername	LDAP directory service account user name
ldapPassword	LDAP directory service account password
description	Description of the LDAP configuration
lockoutDuration	Specifies how long a lookout should last. Set to 0 to use the LDAP directories value

10.2 GET ldap/{id}

Get a LDAP configuration

URL: `https://www.bypass.no/bpcode/merchants/{merchant-id}/ldap/{id}.{json,xml}`

Response

JSON example:

```
{
  "appIdAttrName": "",
  "description": "LDAPBPLAB01",
  "enabled": true,
  "ldapConfigId": 4,
  "ldapPassword": "xxxxx",
  "ldapPaths": [
    {
      "ldapId": 4,
      "ldapPathId": 8,
      "priority": 1,
      "searchString": "basedn[ou=Users,ou=BPCode,dc=BPLAB01,dc=local(*)],filter[samAccountName=#USER#]"
    }
  ],
  "ldapUrl": "ldap://192.168.171.15",
  "ldapUsername": "bpc_svc_ldap",
  "lockoutDuration": 0,
  "merchantId": 0,
  "phoneNumberAttrName": "phoneNumber;mobile",
  "priority": 2,
  "securityRequirement": ""
}
```

XML example:

```
1. <list>
2.   <LdapModel>
3.     <ldapConfigId class="int">4</ldapConfigId>
4.     <ldapUrl>ldap://192.168.171.15</ldapUrl>
5.     <ldapUsername>bpc_svc_ldap</ldapUsername>
6.     <ldapPassword>xxxxx</ldapPassword>
7.     <phoneNumberAttrName>phoneNumber;mobile</phoneNumberAttrName>
8.     <priority class="big-decimal">2</priority>
9.     <description>LDAP BPLAB01</description>
10.    <enabled>true</enabled>
11.    <lockoutDuration>0</lockoutDuration>
12.    <ldapPaths class="list">
13.      <LdapPathModel>
14.        <ldapConfigId class="int">4</ldapConfigId>
15.        <ldapPathId class="int">8</ldapPathId>
16.        <priority class="big-decimal">1</priority>
```

```

17.     <searchString>basedn[ou=Users,ou=BPCode,dc=BPLAB01,dc=local(*)],filter[samA
      ccountName=#USER#]</searchString>
18.     </LdapPathModel>
19.     </LdapPaths>
20.   </LdapModel>
21. </list>

```

Response fields

Fields	Description
ldapConfigId	LDAP configuration id
ldapUrl	URI to the LDAP directory
ldapPaths	Array of LDAP paths, see ldap-path section
priority	Priority of the LDAP configuration
merchantId	Deprecated
appIdAttrName	Attribute in LDAP directory that contains the user's appId
phoneNumberAttrName	Attribute in LDAP directory that contains the user's phone number
ldapUsername	LDAP directory service account user name
ldapPassword	LDAP directory service account password
description	Description of the LDAP configuration
lockoutDuration	Specifies how long a lookout should last. Set to 0 to use the LDAP directories value

10.3 POST ldap

Create new LDAP configuration.

URL: <https://www.bypass.no/bpcode/merchants/{merchant-id}/ldap.{json, xml}>

10.4 PUT ldap/{id}

Update an existing LDAP configuration.

URL: <https://www.bypass.no/bpcode/merchants/{merchant-id}/ldap/{id}.{json, xml}>

10.5 DELETE ldap/{id}

Delete a LDAP configuration

URL: <https://www.bypass.no/bpcode/merchants/{merchant-id}/ldap/{id}.{json, xml}>

10.6 GET ldap/{id}/up

Swap priority of this LDAP configuration with the one before, e.g. if this LDAP configuration has priority = 2 it will have priority = 1 after this request and the one which previously had priority = 1 will have priority = 2.

URL: <https://www.bypass.no/bpcode/merchants/{merchant-id}/ldap/{id}/up.{json, xml}>

10.7 GET ldap/{id}/down

Swap priority of this LDAP configuration with the one after, e.g. if this LDAP configuration has priority = 2 it will have priority = 3 after this request and the one which previously had priority = 3 will have

priority = 2.

URL: `https://www.buypass.no/bpcode/merchants/{merchant-id}/ldap/{id}/down.{json, xml}`

10.8 GET ldap/{id}/ldap-path

Get all LDAP paths for a LDAP configuration

URL: `https://www.buypass.no/bpcode/merchants/{merchant-id}/ldap/{id}/ldap-path.{json, xml}`

Response

JSON example:

```
[
  {
    "ldapId":4,
    "ldapPathId":8,
    "priority":1,
    "searchString":"basedn[ou=Users,ou=BPCode,dc=BPLAB01,dc=local(*)],filter[samAccountName=#USER#]"
  }
]
```

XML example:

```
1. <list>
2.   <LdapPathModel>
3.     <ldapConfigId class="int">4</ldapConfigId>
4.     <ldapPathId class="int">8</ldapPathId>
5.     <priority class="big-decimal">1</priority>
6.     <searchString>basedn[ou=Users,ou=BPCode,dc=BPLAB01,dc=local(*)],filter[samAccountName=#USER#]</searchString>
7.   </LdapPathModel>
8. </list>
```

Response fields

Fields	Description
ldapConfigId	Id of the LDAP configuration that this LDAP path belongs to
ldapPathId	Id of this LDAP path
priority	Priority of this LDAP path
searchString	Search string for this LDAP path

10.9 GET ldap/{id}/ldap-path/{priority} ³

Get a LDAP path configuration.

URL: `https://www.buypass.no/bpcode/merchants/{merchant-id}/ldap/{id}/ldap-path/{priority}.{json, xml}`

Response

JSON example:

```
{
  "ldapId":4,
  "ldapPathId":8,
  "priority":1,
  "searchString":"basedn[ou=Users,ou=BPCode,dc=BPLAB01,dc=local(*)],filter[samAccountName=#USER#]"
}
```

³ Available from release 7.27.X.X, June 2014

```
untName=#USER#]"
}
```

XML example:

```
1. <LdapPathModel>
2.   <ldapConfigId class="int">4</ldapConfigId>
3.   <ldapPathId class="int">8</ldapPathId>
4.   <priority class="big-decimal">1</priority>
5.   <searchString>basedn[ou=Users,ou=BPCode,dc=BPLAB01,dc=local(*)],filter[samAccountName=#USER#]</searchString>
6. </LdapPathModel>
```

Response fields

Fields	Description
ldapConfigId	Id of the LDAP configuration that this LDAP path belongs to
ldapPathId	Id of this LDAP path
priority	Priority of this LDAP path
searchString	Search string for this LDAP path

10.10 POST ldap/{id}/ldap-path

Create new LDAP path for a LDAP configuration.

URL: <https://www.bypass.no/bpcode/merchants/{merchant-id}/ldap/{id}/ldap-path.{json, xml}>

10.11 PUT ldap/{id}/ldap-path/{priority}

Update an existing LDAP path for a LDAP configuration.

URL: <https://www.bypass.no/bpcode/merchants/{merchant-id}/ldap/{id}/ldap-path/{priority}.{json, xml}>

10.12 DELETE ldap/{id}/ldap-path/{priority}

Delete a LDAP path for a LDAP configuration.

URL: <https://www.bypass.no/bpcode/merchants/{merchant-id}/ldap/{id}/ldap-path/{priority}.{json, xml}>

10.13 GET cache

Empty all users' LDAP cache.

URL: <https://www.bypass.no/bpcode/merchants/{merchant-id}/cache.{json, xml}>

11 Radius

11.1 GET radius

Get all Radius configurations.

URL: <https://www.bypass.no/bpcode/merchants/{merchant-id}/radius.{json, xml}>

Response

JSON example:

```
[
  {
```

```

    "allowOfflineOtp":true,
    "allowOnlyProbe":false,
    "chosenAttributes":[
      {
        "def":"NAS_IDENTIFIER",
        "value":"adfsproxy"
      }
    ],
    "description":"ADFSProxy",
    "externalRadius":[
    ],
    "ipAddress":"192.168.173.44",
    "ldaps":[
      {
        "appIdAttrName":"","
        "description":"LDAPBPLAB01",
        "enabled":true,
        "ldapConfigId":7,
        "ldapPassword":"xxxxx",
        "ldapPaths":[
          {
            "ldapId":7,
            "ldapPathId":8,
            "priority":3,
            "searchString":"basedn[ou=Users3,ou=BPCode,dc=BPLAB01,dc=local](*)"
          }
        ],filter[samAccountName=#USER#]"
      }
    ],
    "ldapUrl":"ldap://192.168.171.15",
    "ldapUsername":"bpc_svc_ldap",
    "lockoutDuration":0,
    "merchantId":0,
    "phoneNumberAttrName":"phoneNumber;mobile",
    "priority":2,
    "securityRequirement":""
  }
],
"merchantId":6007,
"nasIdentifier":"","
"nasIpAddress":"","
"probePassword":"","
"probeUsername":"","
"probingEnabled":false,
"radiusConfigId":24,
"sharedSecret":"secret",
"tunnelGroupName":""
}
]

```

XML example:

```

1. <list>
2. <Radius>
3. <merchantId class="long">6007</merchantId>
4. <radiusConfigId class="int">22</radiusConfigId>
5. <ipAddress>192.168.173.44</ipAddress>
6. <sharedSecret>secretkey</sharedSecret>
7. <nasIdentifier>adfsproxy</nasIdentifier>
8. <description>ADFS Proxy i lab</description>
9. <probingEnabled>>false</probingEnabled>
10. <allowOnlyProbe>>false</allowOnlyProbe>
11. <allowOfflineOtp>>true</allowOfflineOtp>
12. <ldaps class="list">

```

```

13. <Ldap>
14. <ldapConfigId class="int">6</ldapConfigId>
15. <ldapUrl>ldap://192.168.171.15</ldapUrl>
16. <ldapUsername>bpc_svc_ldap</ldapUsername>
17. <ldapPassword>Test1234</ldapPassword>
18. <phoneNumberAttrName>phoneNumber;mobile</phoneNumberAttrName>
19. <priority class="int">2</priority>
20. <description>LDAP BPLAB01</description>
21. <enabled>>true</enabled>
22. <lockoutDuration>0</lockoutDuration>
23. <ldapPaths class="list">
24. <LdapPath>
25. <ldapConfigId class="int">6</ldapConfigId>
26. <ldapPathId class="int">7</ldapPathId>
27. <priority class="long">3</priority>
28. <searchString>
29. basedn[ou=Users3,ou=BPCode,dc=BPLAB01,dc=local(*)],filter[samAccountName=#USER#]</se
archString>
30. </LdapPath>
31. </ldapPaths>
32. </Ldap>
33. </ldaps>
34. <externalRadius class="list"/>
35. <chosenAttributes class="list">
36. <RadiusAttribute>
37. <def>NAS_IDENTIFIER</def>
38. <value>adfsproxy</value>
39. </RadiusAttribute>
40. </chosenAttributes>
41. </Radius>
42. </list>

```

Response fields

Fields	Description
merchantId	Merchant's id
radiusConfigId	Radius configuration id
ipAddress	IP-address of the Radius client
sharedSecret	Pre-shared secret for secure Radius communication
nasIpAddress	NAS IP-address
nasIdentifier	NAS identifier
tunnelGroupName	Tunnel group name
description	Description of the Radius configuration
probingEnabled	Decides if probing is enabled
allowOnlyProbe	Decides if only probing should be allowed for this Radius configuration
allowOfflineOtp	Decides if offline OTPs should be allowed
ldaps	Array of LDAP configurations that this Radius configuration uses for user lookup, see LDAP section
externalRadius	Array of external Radius configurations that this Radius configuration can proxy requests to, see external Radius
chosenAttributes	Array of Radius attributes. A Radius attribute has a def field (NAS_IDENTIFIER, NAS_IP_ADDRESS or TUNNEL_GROUP_NAME) and a value field.

11.2 GET radius/{id}

Get a Radius configuration.

URL: `https://www.buypass.no/bpcode/merchants/{merchant-id}/radius/{id}.{json, xml}`

Response

JSON example:

```
{
  "allowOfflineOtp":true,
  "allowOnlyProbe":false,
  "chosenAttributes":[
    {
      "def":"NAS_IDENTIFIER",
      "value":"adsproxy"
    }
  ],
  "description":"ADFSProxy",
  "externalRadius":[
  ],
  "ipAddress":"192.168.173.44",
  "ldaps":[
    {
      "appIdAttrName":"","
      "description":"LDAPBPLAB01",
      "enabled":true,
      "ldapConfigId":7,
      "ldapPassword":"xxxxx",
      "ldapPaths":[
        {
          "ldapId":7,
          "ldapPathId":8,
          "priority":3,
          "searchString":"basedn[ou=Users3,ou=BPCode,dc=BPLAB01,dc=local(*)],filter[samAccountName=#USER#]"
        }
      ],
      "ldapUrl":"ldap://192.168.171.15",
      "ldapUsername":"bpc_svc_ldap",
      "lockoutDuration":0,
      "merchantId":0,
      "phoneNumberAttrName":"phoneNumber;mobile",
      "priority":2,
      "securityRequirement":""
    }
  ],
  "merchantId":6007,
  "nasIdentifier":"","
  "nasIpAddress":"","
  "probePassword":"","
  "probeUname":"","
  "probingEnabled":false,
  "radiusConfigId":24,
  "sharedSecret":"secret",
  "tunnelGroupName":""
}
```

XML example:

```
1. <Radius>
2. <merchantId class="long">6007</merchantId>
3. <radiusConfigId class="int">22</radiusConfigId>
4. <ipAddress>192.168.173.44</ipAddress>
```

```

5. <sharedSecret>secretkey</sharedSecret>
6. <nasIdentifier>adfsproxy</nasIdentifier>
7. <description>ADFS Proxy i lab</description>
8. <probingEnabled>false</probingEnabled>
9. <allowOnlyProbe>false</allowOnlyProbe>
10. <allowOfflineOtp>true</allowOfflineOtp>
11. <ldaps class="list">
12.   <Ldap>
13.     <ldapConfigId class="int">6</ldapConfigId>
14.     <ldapUrl>ldap://192.168.171.15</ldapUrl>
15.     <ldapUsername>bpc_svc_ldap</ldapUsername>
16.     <ldapPassword>Test1234</ldapPassword>
17.     <phoneNumberAttrName>phoneNumber;mobile</phoneNumberAttrName>
18.     <priority class="int">2</priority>
19.     <description>LDAP BPLAB01</description>
20.     <enabled>true</enabled>
21.     <lockoutDuration>0</lockoutDuration>
22.     <ldapPaths class="list">
23.       <LdapPath>
24.         <ldapConfigId class="int">6</ldapConfigId>
25.         <ldapPathId class="int">7</ldapPathId>
26.         <priority class="long">3</priority>
27.         <searchString>
28. basedn[ou=Users3,ou=BPCode,dc=BPLAB01,dc=local(*)],filter[samAccountName=#USER#]</se
archString>
29.       </LdapPath>
30.     </ldapPaths>
31.   </Ldap>
32. </ldaps>
33. <externalRadius class="list"/>
34. <chosenAttributes class="list">
35.   <RadiusAttribute>
36.     <def>NAS_IDENTIFIER</def>
37.     <value>adfsproxy</value>
38.   </RadiusAttribute>
39. </chosenAttributes>
40. </Radius>

```

Response fields

Fields	Description
merchantId	Merchant's id
radiusConfigId	Radius configuration id
ipAddress	IP-address of the Radius client
sharedSecret	Pre-shared secret for secure Radius communication
nasIpAddress	NAS IP-address
nasIdentifier	NAS identifier
tunnelGroupName	Tunnel group name
description	Description of the Radius configuration
probingEnabled	Decides if probing is enabled
allowOnlyProbe	Decides if only probing should be allowed for this Radius configuration
allowOfflineOtp	Decides if offline OTPs should be allowed
ldaps	Array of LDAP configurations that this Radius configuration uses for user lookup, see LDAP section
externalRadius	Array of external Radius configurations that this Radius configuration can

	proxy requests to, see external Radius
chosenAttributes	Array of Radius attributes. A Radius attribute has a def field (NAS_IDENTIFIER, NAS_IP_ADDRESS or TUNNEL_GROUP_NAME) and a value field.

11.3 POST radius

Create new Radius configuration.

URL: `https://www.bypass.no/bpcode/merchants/{merchant-id}/radius.{json,xml}[?arg1=val1]`

Parameters

Name	Type	Description
ldapIds	int	Id of LDAP that should be used for lookup by this Radius configuration. Multiple parameters allowed.
chosenAttribute	string;string	Radius attribute [attributeDefinition;value]. Allowed attribute definitions NAS_IDENTIFIER, NAS_IP_ADDRESS, TUNNEL_GROUP_NAME. Multiple parameters allowed with different attribute definitions.
externalRadiusIds	int	Id of external Radius configuration that requests should be proxied to from this Radius configuration. Multiple parameters allowed.

11.4 PUT radius/{id}

Update an existing Radius configuration.

URL: `https://www.bypass.no/bpcode/merchants/{merchant-id}/radius/{id}.{json,xml}[?arg1=val1]`

Parameters

Name	Type	Description
ldapIds	int	Id of LDAP that should be used for lookup by this Radius configuration. Multiple parameters allowed.
chosenAttribute	string;string	Radius attribute [attributeDefinition;value]. Allowed attribute definitions NAS_IDENTIFIER, NAS_IP_ADDRESS, TUNNEL_GROUP_NAME. Multiple parameters allowed with different attribute definitions.
externalRadiusIds	int	Id of external Radius configuration that requests should be proxied to from this Radius configuration. Multiple parameters allowed.

11.5 DELETE radius/{id}

Delete a Radius configuration.

URL: `https://www.bypass.no/bpcode/merchants/{merchant-id}/radius/{id}.{json,xml}`

12 External Radius

12.1 GET external-radius⁴

Get all external Radius configurations.

URL: `https://www.buypass.no/bpcode/merchants/{merchant-id}/external-radius.{json, xml}`

Response

JSON example:

```
[
  {
    "description": "RSA server",
    "externalRadiusId": 3,
    "ipAddress": "192.168.171.12",
    "merchantId": 0,
    "port": 1812,
    "retries": 3,
    "sharedSecret": "xxxxx",
    "timeoutMs": 3000
  }
]
```

XML example:

```
1. <list>
2. <ExternalRadius>
3. <externalRadiusId class="int">3</externalRadiusId>
4. <ipAddress>192.168.171.12</ipAddress>
5. <port>1812</port>
6. <sharedSecret>Passw0rd</sharedSecret>
7. <timeoutMs>3000</timeoutMs>
8. <retries>3</retries>
9. <description>RSA server</description>
10. </ExternalRadius>
11. </list>
```

Response fields

Fields	Description
merchantId	Deprecated
externalRadiusConfigId	Id of external Radius configuration
ipAddress	IP-address to the external Radius
port	Port that the external Radius listens to
sharedSecret	Pre-shared secret of the Radius communication. Has to be the same as on the external Radius server.
timeoutMs	Timeout in milliseconds for each attempt
retries	Number of attempts
description	Description of the external Radius configuration

12.2 GET external-radius/{id}

Get an external Radius configuration.

⁴ Available from release 7.27.X.X, June 2014

URL: `https://www.bypass.no/bpcode/merchants/{merchant-id}/external-radius/{id}.{json, xml}`

Response

JSON example:

```
[
  {
    "description": "RSA server",
    "externalRadiusId": 3,
    "ipAddress": "192.168.171.12",
    "merchantId": 0,
    "port": 1812,
    "retries": 3,
    "sharedSecret": "xxxxx",
    "timeoutMs": 3000
  }
]
```

XML example:

```
1. <ExternalRadius>
2.   <externalRadiusId class="int">3</externalRadiusId>
3.   <ipAddress>192.168.171.12</ipAddress>
4.   <port>1812</port>
5.   <sharedSecret>Passw0rd</sharedSecret>
6.   <timeoutMs>3000</timeoutMs>
7.   <retries>3</retries>
8.   <description>RSA server</description>
9. </ExternalRadius>
```

Response fields

Fields	Description
merchantId	Deprecated
externalRadiusConfigId	Id of external Radius configuration
ipAddress	IP-address to the external Radius
port	Port that the external Radius listens to
sharedSecret	Pre-shared secret of the Radius communication. Has to be the same as on the external Radius server.
timeoutMs	Timeout in milliseconds for each attempt
retries	Number of attempts
description	Description of the external Radius configuration

12.3 POST external-radius

Create new external Radius configuration.

URL: `https://www.bypass.no/bpcode/merchants/{merchant-id}/external-radius.{json, xml}`

12.4 PUT external-radius/{id}

Update an existing external Radius configuration.

URL: `https://www.bypass.no/bpcode/merchants/{merchant-id}/external-radius/{id}.{json, xml}`

12.5 DELETE external-radius/{id}

Delete an external Radius configuration.

URL: `https://www.buypass.no/bpcode/merchants/{merchant-id}/external-radius/{id}.{json, xml}`