BiteAPI

Information for Service Providers

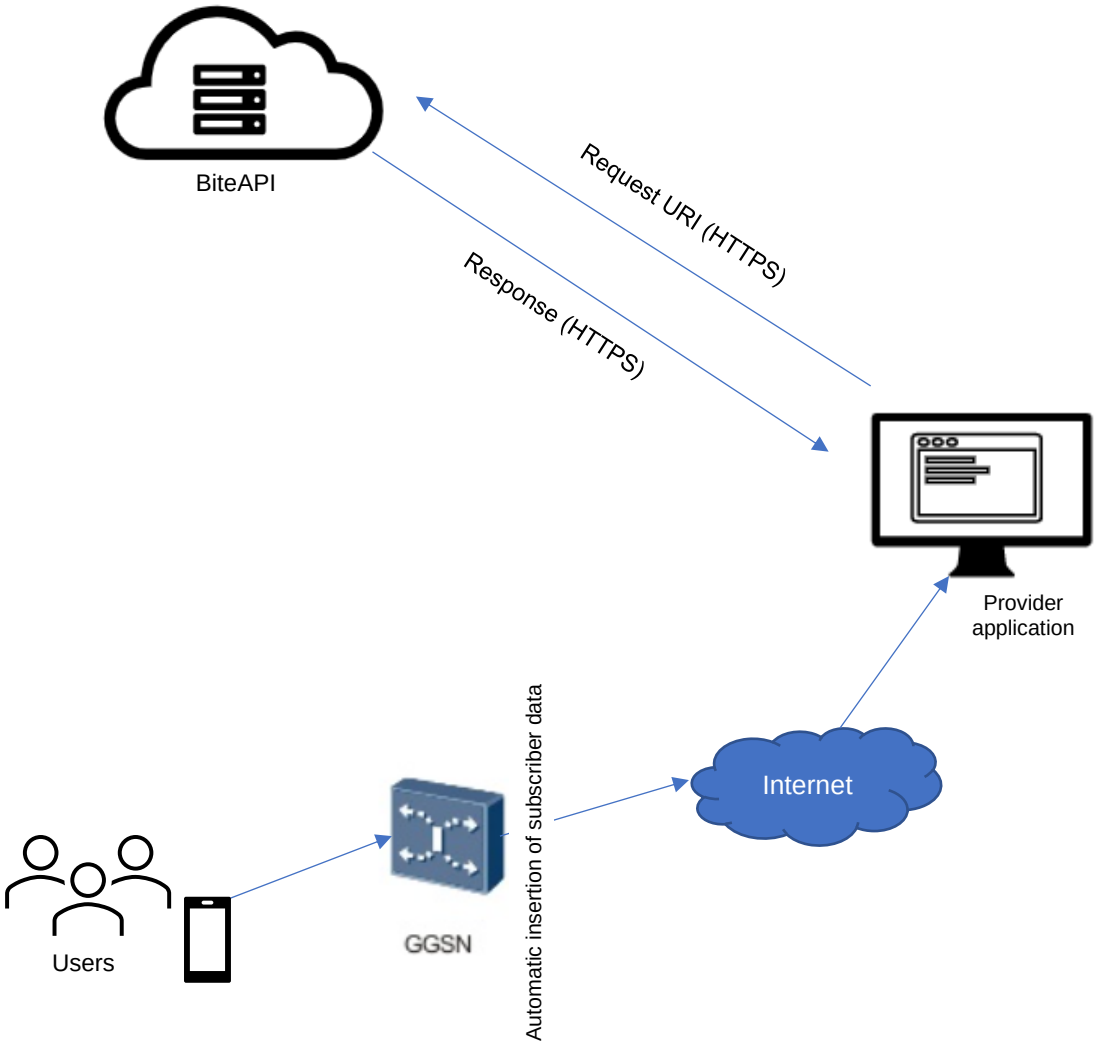
version 4.6, 2023-04-27

Contents

1. What is BiteAPI?

BiteAPI is a HTTPS/REST interface to mobile internet system, which allows providers to perform billing and other functions needed by their premium products. This guide is written for developers writing applications using BiteAPI web services interface. It includes general information and overview of operations which can be performed using its' web services interface.

2. Request Scheme



Few things are important when using the BiteAPI web services interface. First, the provider authentication should be done using OAUTH2 authentication method (recommended). Therefore, the interface expects valid access token within the HTTP headers. The system will not allow requests without this information.

3. Providers

Each content provider (i.e. a legal entity) has his own user name and password which must be used to authenticate to the API system.

4. Services

Each Provider can operate many premium services, for example “Weather reports”, “Monthly service subscription”. A service is something which is understood by the user as a single premium service and has a human understandable description. A service is supplied for a given market. Therefore, running the same service in Lithuania and Latvia still requires setting up two separate services to reflect currency and accounting differences.

5. Actions

Each service is delivered to the user via paid actions, such as “File download” or “Monthly subscription”, or “SMS to the user”. Each action has one or more prices that the end-user can be charged and may have an associated cost for the provider too (in case of sending SMS to the user).

6. System Manager

All configuration of the API including partner account setup, service setup, price setup and other administrative activities are performed by a person responsible of managing the partners services, their actions, and prices. Email: partner_support@bite.lt

7. BiteApi authentication for Providers

Please refer to authentication document for more details (Bite_api_authentification_en.pdf).

8. End-user identification

All end users are identified by their unique information. When the user passes through the Internet Gateway, encrypted customer data becomes available (User-Identity-Forward-msisdn, imei-sv and charging-id). All other persons who are browsing your site or application and who are not BITE network users will not have it. Charging-id is never reused, it is unique to every phone number and person using it. This data is available through special HTTP headers (added when the request passes Bite gateway, automatically for all Bite accounts for whitelisted domains or IP addresses).
Headers that are used in BiteApi:

```
HEADERS:
'User-Identity-Forward-msisdn: k+Gmw9ucvL3zhQ='
'charging-id: 1wUchNNWfwR+Q=='
'imei-sv: cT3Sx0J0r37nbefBWMVmA=='
```

For decrypting encoded headers, please use provided end point in provider API section

Warning: Provider should never use the header information which was acquired outside Bite network. Only those which have been collected from the user request passing the Bite gateway are allowed.

9. Billing

Billing BiteAPI allows partners to perform billing transactions to bill their users. Transaction opening, committing and withdrawal is supported along with plain un transactional billing. Which allows to perform single step billing in cases where transactions are not feasible. To bill, refund or open a billing transaction several input parameters must be supplied. They identify MSISDN number that should be billed, a service and associated account action, as well as price and currency. To open transaction “open-transaction” endpoint is used. It returns an ID of transaction being opened. Having an opened transaction, the partner can now operate within current transaction context and use a “commit-transaction” endpoint in case of successful partner operation or “rollback-transaction” the transaction in case of failure. Both commit and rollback functions use the same transaction ID returned when transaction was opened. There are cases when it is not possible to perform operation or a group of operations within transactional context due to the limitations posed by the nature of these operations.

In such cases “execute-transaction” endpoint should be used. It accepts the same parameters as transactional open-transaction. Each account action is associated with a list of allowed prices (with specified currencies). It is disallowed to perform billing operation with prices outside these values. Prices and actions are managed by system manager.

Detailed billing service description

All HTTPS request examples can be imported to Postman API.

Open transaction

Address	https://biteapi.bite.lt/transactions/open-transaction
Method	POST
Description	Open billing transaction which can be rollbacked or committed later. Two step billing.
Mandatory HTTP request headers	**Authorization** – auth token
Input parameters	**accountId** – customer account identification (integer, mandatory) **serviceId** – service identification (integer, mandatory) **actionId** – action identification (integer, mandatory) **amount** – amount for billing (number, mandatory) **currency** – currency for billing (string, mandatory) (expected value: EUR) **serviceData** – payment description (string, optional)
Return value	**transactionId** – id of opened transaction (integer)

HTTPS request example	`curl --location --request POST 'https://biteapi.bite.lt/transactions/open-transaction' \`
	`--header 'Authorization: Bearer OAUTH_TOKEN' \`
	`--header 'Content-Type: application/json' \`
	`--data-raw '{`
	`"accountId": 123,`
	`"serviceId": 321,`
	`"actionId": 456,`
	`"amount": 0.01,`
	`"currency": "EUR",`
	`"serviceData": "BiteApiTest"`
	`}'`

HTTPS response example	`{`
	`"transactionId": 2933872273`
	`}`

HTTPS error response	HTTP status: 400;
	Response example:
	{
	"code": "-2",
	"message": "System error"
	}

Possible errors	-1035 : Service can not be provided for this customer; -1034 : VAS service is blocked for MSISDN(Account); -1031 : Service id is not valid; -1033 : MSISDN(Account) is not active; -1030 : Account not found; -1101 : Transaction amount is not valid;

	-2 : Service not find or not valid; 1020 : Insufficient customer balance; 1021 : Account is blocked; -10000 : System error (please contact BiteApi system manager regarding this error).

Commit transaction

Address	https://biteapi.bite.lt/transactions/commit-transaction
Method	POST
Description	Commits billing transaction that has been opened before. If transactions that has been opened before will be not committed in several hours, then such records will be automatically withdrawn.
Mandatory HTTP request headers	**Authorization** – auth token
Input parameters	**transactionId** - ID of the created transaction (integer, mandatory)
Return value	status – returns 'OK' if commit operation has been successful (string)

HTTPS request example	`curl --location --request POST 'https://biteapi.bite.lt/transactions/commit-transaction' \`
	`--header 'Authorization: Bearer OAUTH_TOKEN' \`
	`--header 'Content-Type: application/json' \`
	`--data-raw '{`
	`"transactionId": 2952657668`
	`}'`

HTTPS response example	`{`
	`"status": "OK"`
	`}`

Possible errors:	In case of any error HTTPS response will differ and return error code with message.
	Example:
	{
	"code": "-1",
	"message": "System error"
	}

Possible errors	-1003 : Transaction is not opened or already closed; -10000 : System error (please contact BiteApi system manager regarding this error).

Rollback transaction

Address	https://biteapi.bite.lt/transactions/rollback-transaction
Method	POST
Description	Rollbacks billing transaction that has been opened before. If transactions that has been opened before will be not committed/rollbacked in several hours, then such records will be automatically withdrawn.
Mandatory HTTP request headers	**Authorization** – auth token
Input parameters	**transactionId** - ID of the created transaction (integer, mandatory)

Return value	status – returns 'OK' if commit operation has been successfull (string)

HTTPS request example	`curl --location --request POST 'https://biteapi.bite.lt/transactions/rollback-transaction' \`
	`--header 'Authorization: Bearer OAUTH_TOKEN' \`
	`--header 'Content-Type: application/json' \`
	`--data-raw '{`
	`"transactionId": 2952657671`
	`}'`

HTTPS response example	`{`
	`"status": "OK"`
	`}`

HTTPS error response	Example:
	{
	"code":"-1",
	"message":"System error"
	}

Possible errors	-1003 : Transaction is not opened or already closed; -10000 : System error (please contact BiteApi system manager regarding this error).

Execute transaction

Address	https://biteapi.bite.lt/transactions/execute-transaction
Method	POST
Description	One step billing – transaction is opened and closed at the same time.
Mandatory HTTP request headers	**Authorization** – auth token
Input parameters	**accountId** – customer account identification (integer, mandatory) **serviceId** – service identification (integer, mandatory) **actionId** – action identification (integer, mandatory) **amount** – amount for billing (number, mandatory) **currency** – currency for billing (string, mandatory) (expected value: EUR) **serviceData** – payment description (string, optional)
Return value	**transactionId** - Returns the ID of the created transaction (integer)

HTTPS request example	`curl --location --request POST 'https://biteapi.bite.lt/transactions/execute-transaction' \`
	`--header 'Authorization: Bearer OAUTH_TOKEN' \`
	`--header 'Content-Type: application/json' \`
	`--data-raw '{`
	`"accountId": 123,`
	`"serviceId": 321,`
	`"actionId": 456,`
	`"amount": 0.01,`
	`"currency": "EUR",`
	`"serviceData": "BiteApiTest"`
	`}'`

HTTPS response example	`{`
	`"transactionId": 2933872273`
	`}`

HTTPS error response	HTTP status: 400;
	Response example:
	{
	"code": "-2",
	"message": "System error"
	}

Possible errors	-1035 : Service can not be provided for this customer; -1034 : VAS service is blocked for MSISDN(Account); -1031 : Service id is not valid; -1033 : MSISDN(Account) is not active; -1030 : Account not found; -1101 : Transaction amount is not valid; -2 : Service not found or not valid; 1020 : Insufficient customer balance; 1021 : Account is blocked; -10000 : System error (please contact BiteApi system manager regarding this error).

Transaction status

Address	https://biteapi.bite.lt/transactions/transaction-status
Method	POST
Description	Returns transaction status.
Mandatory HTTP request headers	**Authorization** – auth token
Input parameters	**transactionId** – transaction identificator (integer, mandatory)
Return value	status – transaction status (string)
Possible return values	CANCELLED : rollbacked transaction status; IN PROGRESS : transaction status is OPEN, it is not finished yet; COMPLETED : transaction is completed; NOT_FOUND : when transaction id can not be found in the system.

HTTPS request example	`curl --location --request POST 'https://biteapi.bite.lt/transactions/transaction-status' \`
	`--header 'Authorization: Bearer OAUTH_TOKEN' \`
	`--header 'Content-Type: application/json' \`
	`--data-raw '{`
	`"transactionId": 2952657669`
	`}'`

HTTPS response example	{
	"status": "CANCELLED"
	}

Refund transaction

Address	https://biteapi.bite.lt/transactions/refund-transaction
Method	POST
Description	Refunds already closed transaction for customer.

Mandatory HTTP request headers	**Authorization** – auth token
Input parameters	**accountId** – customer account identification (integer, mandatory) **serviceId** – service identification (integer, mandatory) **actionId** – action identification (integer, mandatory) **amount** – amount for billing (number, mandatory) **currency** – currency for billing (string, mandatory) (expected value: EUR) **serviceData** – payment description (string, optional) **relatedTransactionId** – Id of transaction that should be refunded (integer, mandatory)
Return value	**transactionId** – refund transaction id;

HTTPS request example	`curl --location --request POST 'https://biteapi.bite.lt/transactions/refund-transaction' \`
	`--header 'Authorization: Bearer OAUTH_TOKEN' \`
	`--header 'Content-Type: application/json' \`
	`--data-raw '{`
	`"accountId": 123,`
	`"serviceId": 321,`
	`"actionId": 456,`
	`"amount": "0.5",`
	`"currency": "EUR",`
	`"relatedTransactionId": 2952657669,`
	`"serviceData": "asdass"`
	`}'`

HTTPS response example	`{`
	`"transactionId": 2933872273`
	`}`

HTTPS error response	HTTP status: 400;
	Response example:
	{
	"code": "-2",
	"message": "System error"
	}

Possible errors	-1100 : Related transaction id is not found; -1035 : Service can not be provided for this customer; -1034 : VAS service is blocked for MSISDN(Account); -1031 : Service id is not valid; -1033 : MSISDN(Account) is not active; -1030 : Account not found; -1101 : Transaction amount is not valid; -2 : Provider action not found or not valid; 1020 : Insufficient customer balance; 1021 : Account is blocked; -10000 : System error (please contact BiteApi system manager regarding this error).

10. Provider

Providers can get information about their services and functions of those services, also get decoded msisdn number.

Decode header enrichment data

Address	https://biteapi.bite.lt/provider/header-enrichment/decode-headers
Method	GET
Description	Resolves account msisdn data from header.
Mandatory HTTP request headers	**Authorization** – auth token **User-Identity-Forward-msisdn** – encoded MSISDN information that identifies customer (partner gets it from header enrichment service) **Charging-id** – encoded charging id **Imei-sv** – encoded IMEI Example: --header 'User-Identity-Forward-msisdn: k+Gmw9ucvL3zNhhQ=' \ --header 'charging-id: 1wUchNNWfwR1+Q=h=' \ --header 'imei-sv: cT3Sx0J0r37nbefBWMIVhmA==' \
Input parameters	N/A
Return value	**msisdn** – decrypted msisdn from the header (string) **accountId** – customer account identification (string) **chargingId** – charging ID **imei** – serial number of the device (string)

HTTPS request example	`curl --location --request GET 'https://biteapi.bite.lt/provider/header-enrichment/decode-headers' \`
	`--header 'Authorization: Bearer OAUTH_TOKEN' \`
	`--header 'User-Identity-Forward-msisdn: k+Gnx9uVvL3wNxc=' \`
	`--header 'charging-id: 1wUchNNWfwR1+Q==' \`
	`--header 'imei-sv: cT3Sx0J0r37nbefBWMIVmA=='`

HTTPS response example	`{`
	`"msisdn": "37022222222",`
	`"accountId": "123",`
	`"chargingId": "321",`
	`"imei": "3598471072182201"`
	`}`

HTTPS error response	HTTP status: 400;
	Response example:
	{
	"code": "INTERNAL",
	"message": "System error"
	}

Possible errors	

Get services

Address	https://biteapi.bite.lt/provider/services
Method	GET
Description	Return partner's services

Mandatory HTTP request headers	**Authorization** – auth token
Input parameters	N/A
Return value	**services** – list of services (list) **serviceName** – name of the service (string) **description** – description of the service (string) **providerId** – service provider id (integer) **validFrom** – service valid from date (date) **id** – service id (integer) **bsid** – service bsid (integer)
HTTPS request example	`curl --location --request GET 'https://biteapi.bite.lt/provider/services' \ --header 'Authorization: Bearer OAUTH_TOKEN'`
HTTPS response example	`{ "services": [ { "serviceName": "service name", "description": "service description", "providerId": 1211, "validFrom": "2006-09-19", "id": 1233, "bsid": 3164194 } ]`

HTTPS error response	HTTP status: 400;
	Response example:
	{
	"code": "INTERNAL",
	"message": "System error"
	}

Possible errors	

Get functions

Address	https://biteapi.bite.lt/provider/functions/{**serviceId**}
Method	GET
Description	Return a list of actions (functions) available for specific provider's service
Mandatory HTTP request headers	**Authorization** – auth token
Input parameters	**serviceId** – service identification (integer, mandatory);
Return value	**functions** – function list (list) **actionId** – action id of the service (string) **serviced** – id of the service (string) **providerPrice** – provider price (float) **providerCurrency** – provider currency (string) **userPrice** – user price (integer) **userCurrency** – user currency (string) **validFrom** – date that the service is valid from (date) **id** – id of the function (integer) **bsdata** – bddata of the function (string)

HTTPS request example	curl --location -- request GET 'https://biteapi.bite.lt/provider/functions/321 \ --header 'Authorization: Bearer OAUTH_TOKEN' \ --header 'Content-Type: application/json'
HTTPS response example	{ "functions": [{ "actionId": "456", "serviceId": "321", "providerPrice": 0.03, "providerCurrency": "EUR", "userPrice": 0.28, "userCurrency": "EUR", "validFrom": "2009-08-12 09:36:56", "id": 97853, "bsdata": "37111110000" }]
HTTPS error response	HTTP status: 400; Response example: { "code": "INTERNAL", "message": "System error" }
Possible errors	

11. Customer

Customers BiteAPI allows providers to manage customers related data. For example, get account id by msisdn.

Detailed customer service description

Get account data

URL	https://biteapi.bite.lt/customers/accounts?msisdn={**msisdn**}
Method	GET
Description	Get customers account id by MSISDN
Mandatory HTTP request headers	**Authorization** – auth token
Input parameters	**msisdn** – customer account identification (string, mandatory)
Return value	**accountId** – account id of requested msisdn (integer) **accountType** – account type (string) **status** – account status (string)

Possible return values	**accountType:** PREPAID : prepaid account; POSTPAID : postpaid account; MNP : number is migrated to other provider (not Bite client). **status:** ACTIVE : active account, can be charged; BLOCKED, DEACTIVE, FROZEN, INACTIVE, SUSPENDED : inactive account due to some reason, can't be charged.

HTTPS request example	`curl --location --request GET 'https://biteapi.bite.lt/customers/accounts?msisdn=37011111111' \`
	`--header 'Authorization: Bearer OAUTH_TOKEN' \`
	`--header 'Content-Type: application/json'`

HTTPS response example	`{`
	`"accountId": 123,`
	`"accountType": "POSTPAID",`
	`"status": "ACTIVE"`
	`}`

HTTPS error response	HTTP status: 400;
	Response example:
	{
	"code": "INTERNAL",
	"message": "System error"
	}

Possible errors	-1030 : Account not found.

12. Subscription

Subscription BiteAPI allows providers to manage client subscriptions. Providers can subscribe client to their services and define service billing. Also, subscription service allows to check client subscription information, get subscriptions of chosen service and unsubscribe the client from provider service.

Detailed subscription service description

Subscribe

URL	https://biteapi.bite.lt/subscriptions
Method	POST
Description	Subscribes customer account to specified service. Subscription duration is set to provided term value. After that customer is billed based on renewal value. First bill transaction can optionally be provided
Mandatory HTTP request headers	**Authorization** – auth token
Input parameters	**accountId** – customer account identification (integer, mandatory) **serviceId** – service identification (integer, mandatory) **term** – initial subscription term (integer, mandatory). Allowed values are: 1-24 with "H" termUnit, 1-30 with "D" termUnit, 1 with "W" or "M" termUnit **termUnit** - parameter is used to set unit of term field (string, mandatory). Allowed values are: "H" - hour, "D" - day, "W" - week, "M" - month

	renewal – term for renewal (integer, mandatory). Allowed values are: 1-30 with "D" renewalUnit, 1 with "W" or "M" renewalUnit. Set 0 when no renewal should apply **renewalUnit** – parameter is used to set unit of renewal field (string, mandatory). Allowed values are: "D" - day, "W" - week, "M" – month **transaction** – parameter is used when first subscription billing is needed (object, optional) **actionId** – action identification (integer, mandatory) **amount** – amount for billing (number, mandatory) **currency** – currency for billing (string, mandatory) (expected value: EUR) **serviceData** – payment description (string, optional)
Return value	**statusId** (integer) – returns 0 if the operation was successful **transactionId** (integer) – returns executed transaction id if transaction was provided in input parameters, otherwise returns 0

HTTPS request example	`curl --location --request POST 'https://biteapi.bite.lt/subscriptions' \`
	`--header 'Authorization: Bearer OAUTH_TOKEN' \`
	`--header 'Content-Type: application/json' \`
	`--data-raw '{`
	`"serviceId": 321,`
	`"accountId": 123,`
	`"term": 1,`
	`"termUnit": "D",`
	`"renewal": 1,`
	`"renewalUnit": "D",`
	`"transaction": {`
	`"actionId": 11,`
	`"amount": 1.00,`
	`"currency": "EUR",`
	`"serviceData": "BiteApitest"`
	`}`
	`}'`

HTTPS response example	`{`
	`"statusId": 0,`
	`"transactionId": 2933872273`
	`}`

HTTPS error response	HTTP status: 400;
	Response example:
	{
	"code": "INTERNAL",
	"message": "System error"
	}

Possible errors	

Unsubscribe

URL	https://biteapi.bite.lt/subscriptions/{**serviceId**}/{**accountId**}
Method	DELETE
Description	Unsubscribes customer account from specified service
Mandatory HTTP request headers	**Authorization** – auth token

Input parameters	**accountId** – customer account identification (integer, mandatory) **serviceId** – service identification (integer, mandatory);
Return value	**statusId** (integer) – returns 0 if the operation was successful.

HTTPS request example	`curl --location --`
	`request DELETE 'http://biteapi.bite.lt/subscriptions/321/123' \`
	`--header 'accept: application/json'`

HTTPS response example	`{`
	`"statusId": 0`
	`}`

HTTPS error response	HTTP status: 400;
	Response example:
	{
	"code": "INTERNAL",
	"message": "System error"
	}

Possible errors	

Check status

URL	https://biteapi.bite.lt/subscriptions/{**serviceId**}/{**accountId**}
Method	GET
Description	Return subscription status
Mandatory HTTP request headers	**Authorization** – auth token
Input parameters	**accountId** – customer account identification (integer, mandatory) **serviceId** – service identification (integer, mandatory)
Return value	**status** (string) – status of the subscription **startDate** (string) – subscription start date **endDate** (string) – subscription end date **renewal** (string) – subscription renewal status

HTTPS request example	`curl --location --`
	`request GET 'https://biteapi.bite.lt/subscriptions/321/123' \`
	`--header 'Authorization: Bearer OAUTH_TOKEN' \`
	`--header 'Content-Type: application/json'`

HTTPS response example	`{`
	`"status": "ACTIVE",`
	`"startDate": "2021-08-26 15:50:43",`
	`"endDate": "2021-08-27 15:50:43",`
	`"renewal": "N"`
	`}`

HTTPS error response	HTTP status: 400;
	Response example:
	{
	"code": "INTERNAL",
	"message": "System error"
	}

Possible errors	

Get subscriptions

URL	https://biteapi.bite.lt/subscriptions/{**serviceId**}
Method	GET
Description	Return service subscriptions.
Mandatory HTTP request headers	**Authorization** – auth token
Input parameters	**serviceId** – service identification (integer, mandatory);
Return value	**subscriptions** (list) – list of service subscriptions **accountId** – customer account identification (integer, mandatory) **languageCode** (string) – subscriber language code **msisdn** (string) – subscriber MSISDN **startDate** (string) – start date of the subscription **endDate** (string) – end date of the subscription **status** (string) – subscription status **renewal** (string) – subscription renewal status

HTTPS request example	`curl --location --`
	`request GET 'https://biteapi.bite.lt/subscriptions/321' \`
	`--header 'Authorization: Bearer OAUTH_TOKEN' \`
	`--header 'Content-Type: application/json'`

HTTPS response example	`{`
	`"subscriptions": [`
	`{`
	`"accountId": 123,`
	`"languageCode": "LT",`
	`"msisdn": "3706…",`
	`"startDate": "2021-11-29 15:50:43",`
	`"endDate": "2021-12-06 15:50:43",`
	`"status": "ACTIVE",`
	`"renewal": "W"`
	`}`
	`]`
	`}`

HTTPS error response	HTTP status: 400;
	Response example:
	{
	"code": "INTERNAL",
	"message": "System error"
	}

Possible errors	