

application.client.java.v1.directory.client.api

## Class ClientApi

- [java.lang.Object](#)
  - [application.client.java.v1.ApiBase](#)
    - application.client.java.v1.directory.client.api.ClientApi

---

```
public class ClientApi
extends ApiBase
```

Client Api  
Methods for client management

PATH: [{api}](#)/v1/directory/client

- **Constructor Summary**

### Constructors

Constructor and Description

[ClientApi\(\)](#)

- **Method Summary**

All Methods  
Modifier and Type

Method and Description

static `ActivateClientResponse`

`activateClient(ApiContext ctx, ActivateClientRequest request)`

Activate Client  
Activate a Client in the system.

static `AddClientContractsResponse`

`addClientContracts(ApiContext ctx, AddClientContractsRequest request)`

Add Client Contracts  
Add Client Contracts

PATH: `{api}/v1/directory/client/addClientContracts`

TYPE: AUTHENTICATION

PERMISSIONS:

mandatory: `[directory.client.entity.read.child, directory.client.entity.update, directory.client.contracts.read, directory.client.contracts.write]`

conditional: `[]`

static `AddClientProductsResponse`

`addClientProducts(ApiContext ctx, AddClientProductsRequest request)`

Add Client Products  
Add Client Products

PATH: `{api}/v1/directory/client/addClientProducts`

TYPE: AUTHENTICATION

PERMISSIONS:

mandatory: `[directory.client.entity.read.child, directory.client.entity.update, directory.client.products.read, directory.client.products.write]`

conditional: `[]`

static `CreateClientResponse`

`createClient(ApiContext ctx, CreateClientRequest request)`

Create Client  
Creates a Client in the system.

static `DeactivateClientResponse`

`deactivateClient(ApiContext ctx, DeactivateClientRequest request)`

Deactivate Client  
Deactivate a Client in the system.

static `DeleteClientResponse`

`deleteClient(ApiContext ctx, DeleteClientRequest request)`

Delete Client  
Deletes a Client in the system.

static `DeleteClientWebhookResponse`

`deleteClientWebhook(ApiContext ctx, DeleteClientWebhookRequest request)`

Delete webhook for a client  
Delete webhook for a client

PATH: `{api}/v1/directory/client/deleteClientWebhook`  
TYPE: AUTHENTICATION  
PERMISSIONS:  
 mandatory: `[directory.client.entity.read.self, directory.client.webhook.delete]`  
 conditional: `[]`

`static DisableClientResponse` `disableClient(ApiContext ctx, DisableClientRequest request)`  
Disable Client  
Disables a Client in the system.

`static EnableClientResponse` `enableClient(ApiContext ctx, EnableClientRequest request)`  
Enable Client  
Enables a Client in the system.

`static FindClientsResponse` `findClients(ApiContext ctx, FindClientsRequest request)`  
Find Clients  
Searches for Clients in the system, according to the filter parameters.

`static GetChildrenResponse` `getChildren(ApiContext ctx, GetChildrenRequest request)`  
Get Children  
Get client children

PATH: `{api}/v1/directory/client/getChildren`  
TYPE: AUTHENTICATION  
PERMISSIONS:  
 mandatory: `[directory.client.entity.read.child]`  
 conditional: `[directory.client.products.read, directory.client.contracts.read]`

`static GetClientResponse` `getClient(ApiContext ctx, GetClientRequest request)`  
Get Client  
Get clients

PATH: `{api}/v1/directory/client/getClient`  
TYPE: AUTHENTICATION  
PERMISSIONS:

mandatory: []  
conditional: [directory.client.entity.read.self, directory.client.entity.read.child, directory.client.products.read]

**static GetClientsResponse**

**getClients(ApiContext ctx, GetClientsRequest request)**  
Get Clients  
Get clients

PATH: {api}/v1/directory/client/getClients

TYPE: AUTHENTICATION

PERMISSIONS:

mandatory: []

conditional: [directory.client.entity.read.self, directory.client.entity.read.child, directory.client.products.read]

**static GetClientsProductsResponse**

**getClientsProducts(ApiContext ctx, GetClientsProductsRequest request)**  
Get Clients Products  
Get Clients Products

PATH: {api}/v1/directory/client/getClientsProducts

TYPE: AUTHENTICATION

PERMISSIONS:

mandatory: [directory.client.products.read]

conditional: [directory.client.entity.read.child, directory.client.entity.read.self]

**static GetClientWebhookResponse**

**getClientWebhook(ApiContext ctx, GetClientWebhookRequest request)**  
Get webhook for a client  
Get webhook for a client

PATH: {api}/v1/directory/client/getClientWebhook

TYPE: AUTHENTICATION

PERMISSIONS:

mandatory: [directory.client.entity.read.self, directory.client.webhook.read]

conditional: []

**static PathClientsResponse**

**pathClients(ApiContext ctx, PathClientsRequest request)**  
Path Clients  
Get clients up to root

PATH: {api}/v1/directory/client/pathClients

TYPE: AUTHENTICATION

PERMISSIONS:  
mandatory: []  
conditional: [directory.client.entity.read.self, directory.client.entity.read.child, directory.client.products.read]

static PathsClientsResponse

pathsClients(ApiContext ctx, PathsClientsRequest request)  
Paths Clients  
Get clients up to root

PATH: {api}/v1/directory/client/pathsClients

TYPE: AUTHENTICATION

PERMISSIONS:  
mandatory: []  
conditional: [directory.client.entity.read.self, directory.client.entity.read.child, directory.client.products.read]

static RemoveClientContractsResponse

removeClientContracts(ApiContext ctx, RemoveClientContractsRequest request)  
Remove Client Contracts  
Remove Client Contracts

PATH: {api}/v1/directory/client/removeClientContracts

TYPE: AUTHENTICATION

PERMISSIONS:  
mandatory: [directory.client.entity.read.child, directory.client.entity.update, directory.client.contracts.read, directory.client.contracts.write]  
conditional: []

static RemoveClientProductsResponse

removeClientProducts(ApiContext ctx, RemoveClientProductsRequest request)  
Remove Client Products  
Remove Client Products

PATH: {api}/v1/directory/client/removeClientProducts

TYPE: AUTHENTICATION

PERMISSIONS:  
mandatory: [directory.client.entity.read.child, directory.client.entity.update, directory.client.products.read, directory.client.products.write]  
conditional: []

static RenewClientWebhookKeyResponse

renewClientWebhookKey(ApiContext ctx, RenewClientWebhookKeyRequest request)  
Renew webhook key for webhook  
Set webhook for webhook

PATH: {api}/v1/directory/client/renewClientWebhookKey  
TYPE: AUTHENTICATION  
PERMISSIONS:  
 mandatory: [directory.client.entity.read.self, directory.client.webhook.key.renew]  
 conditional: []

static SetClientProductsResponse

setClientProducts(ApiContext ctx, SetClientProductsRequest request)  
Set Client Products  
Set Client Products

PATH: {api}/v1/directory/client/setClientProducts  
TYPE: AUTHENTICATION  
PERMISSIONS:  
 mandatory: [directory.client.entity.read.child, directory.client.entity.update, directory.client.products.read, directory.client.products.write]  
 conditional: []

static SetClientRisksResponse

setClientRisks(ApiContext ctx, SetClientRisksRequest request)  
Set Client Risk Management  
Set Client Risk Management

PATH: {api}/v1/directory/client/setClientRisks  
TYPE: AUTHENTICATION  
PERMISSIONS:  
 mandatory: [directory.client.entity.read.child, directory.client.risks.write, directory.client.risks.read, directory.client.entity.update]  
 conditional: []

static SetClientSchemeResponse

setClientScheme(ApiContext ctx, SetClientSchemeRequest request)  
Set Client Scheme  
Set Client Scheme

PATH: {api}/v1/directory/client/setClientScheme  
TYPE: AUTHENTICATION  
PERMISSIONS:  
 mandatory: [directory.client.entity.read.child, directory.client.scheme.write]  
 conditional: [directory.client.products.read]

<p><code>static SetClientWebhookResponse</code></p>	<p><code>setClientWebhook(ApiContext ctx, SetClientWebhookRequest request)</code>  Set webhook for a Client  Set webhook for a Client</p> <p>PATH: <code>{api}/v1/directory/client/setClientWebhook</code>  TYPE: AUTHENTICATION  PERMISSIONS:  mandatory: <code>[directory.client.entity.read.self, directory.client.webhook.write]</code>  conditional: <code>[]</code></p>
<p><code>static UndeleteClientResponse</code></p>	<p><code>undeleteClient(ApiContext ctx, UndeleteClientRequest request)</code>  Undelete Client  Undeletes a Client in the system.</p>
<p><code>static UpdateClientResponse</code></p>	<p><code>updateClient(ApiContext ctx, UpdateClientRequest request)</code>  Update Client  Updates a Client in the system.</p>
<p><code>static UpdateClientAddressResponse</code></p> <p><code>updateClientAddress(ApiContext ctx, UpdateClientAddressRequest request)</code>  update Client Address  update Client Address</p> <p>PATH: <code>{api}/v1/directory/client/updateClientAddress</code>  TYPE: AUTHENTICATION  PERMISSIONS:  mandatory: <code>[directory.client.entity.read.child, directory.client.entity.update]</code>  conditional: <code>[directory.client.contracts.read, directory.client.products.read]</code></p>	

- Methods inherited from class `application.client.java.v1.ApiBase`

`process`, `processGet`

- Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

•

- Constructor Detail

- ClientApi

public ClientApi()

- Method Detail

- getClients

- public static [GetClientsResponse](#) getClients([ApiContext](#) ctx,  
• [GetClientsRequest](#) request)  
throws [ApiException](#)

Get Clients  
Get clients

PATH: {api}/v1/directory/client/getClients

TYPE: AUTHENTICATION

PERMISSIONS:

mandatory: []

conditional: [directory.client.entity.read.self, directory.client.entity.read.child, directory.client.products.read]

Parameters:

ctx - Context

request - GetClientsRequest

Returns:

[GetClientsResponse](#)

Throws:

[ApiException](#)

- getClient

- public static [GetClientResponse](#) getClient([ApiContext](#) ctx,  
• [GetClientRequest](#) request)



throws [ApiException](#)

Get Client  
Get clients

PATH: [{api}/v1/directory/client/getClient](#)

TYPE: AUTHENTICATION

PERMISSIONS:

mandatory: []

conditional: [directory.client.entity.read.self, directory.client.entity.read.child, directory.client.products.read]

Parameters:

ctx - Context

request - [GetClientRequest](#)

Returns:

[GetClientResponse](#)

Throws:

[ApiException](#)

- updateClient

- public static [UpdateClientResponse](#) updateClient([ApiContext](#) ctx,  
• [UpdateClientRequest](#) request)  
throws [ApiException](#)

Update Client

Updates a Client in the system.

PATH: [{api}/v1/directory/client/updateClient](#)

TYPE: AUTHENTICATION

PERMISSIONS:

mandatory: [directory.client.entity.read.child, directory.client.entity.update]

conditional: [directory.client.products.read]

Parameters:

ctx - Context

request - [UpdateClientRequest](#)

Returns:

[UpdateClientResponse](#)

Throws:

[ApiException](#)

- **disableClient**

- public static [DisableClientResponse](#) disableClient([ApiContext](#) ctx,  
• [DisableClientRequest](#) request)  
throws [ApiException](#)

Disable Client

Disables a Client in the system.

PATH: [{api}/v1/directory/client/disableClient](#)

TYPE: AUTHENTICATION

PERMISSIONS:

mandatory: [directory.client.entity.read.child, directory.client.entity.disable]

conditional: [directory.client.products.read]

Parameters:

ctx - Context

request - [DisableClientRequest](#)

Returns:

[DisableClientResponse](#)

Throws:

[ApiException](#)

- **enableClient**

- public static [EnableClientResponse](#) enableClient([ApiContext](#) ctx,  
• [EnableClientRequest](#) request)  
throws [ApiException](#)

Enable Client

Enables a Client in the system.

PATH: [{api}/v1/directory/client/enableClient](#)

TYPE: AUTHENTICATION

PERMISSIONS:

mandatory: [directory.client.entity.read.child, directory.client.entity.enable]

conditional: [directory.client.products.read, directory.client.contracts.read]

Parameters:

ctx - Context

request - [EnableClientRequest](#)

Returns:

[EnableClientResponse](#)

Throws:

[ApiException](#)

- deleteClient

- public static [DeleteClientResponse](#) deleteClient([ApiContext](#) ctx,
- [DeleteClientRequest](#) request)  
throws [ApiException](#)

Delete Client

Deletes a Client in the system.

PATH: [{api}/v1/directory/client/deleteClient](#)

TYPE: AUTHENTICATION

PERMISSIONS:

mandatory: [directory.client.entity.read.child, directory.client.entity.delete]

conditional: [directory.client.products.read]

Parameters:

ctx - Context

request - [DeleteClientRequest](#)

Returns:

[DeleteClientResponse](#)

Throws:

[ApiException](#)

- undeleteClient

- public static [UndeleteClientResponse](#) undeleteClient([ApiContext](#) ctx,  
• [UndeleteClientRequest](#) request)  
throws [ApiException](#)

Undelete Client

Undeletes a Client in the system.

PATH: [{api}/v1/directory/client/undeleteClient](#)

TYPE: AUTHENTICATION

PERMISSIONS:

mandatory: [directory.client.entity.read.child, directory.client.entity.undelete]

conditional: [directory.client.products.read]

Parameters:

ctx - Context

request - UndeleteClientRequest

Returns:

[UndeleteClientResponse](#)

Throws:

[ApiException](#)

- findClients

- public static [FindClientsResponse](#) findClients([ApiContext](#) ctx,  
• [FindClientsRequest](#) request)  
throws [ApiException](#)

Find Clients

Searches for Clients in the system, according to the filter parameters.

PATH: [{api}/v1/directory/client/findClients](#)

TYPE: AUTHENTICATION

PERMISSIONS:

mandatory: [directory.client.entity.read.child]  
conditional: [directory.client.entity.read.self]

Parameters:

ctx - Context

request - FindClientsRequest

Returns:

FindClientsResponse

Throws:

[ApiException](#)

- pathClients

- public static [PathClientsResponse](#) pathClients([ApiContext](#) ctx,  
• [PathClientsRequest](#) request)  
throws [ApiException](#)

Path Clients

Get clients up to root

PATH: {api}/v1/directory/client/pathClients

TYPE: AUTHENTICATION

PERMISSIONS:

mandatory: []

conditional: [directory.client.entity.read.self, directory.client.entity.read.child, directory.client.products.read]

Parameters:

ctx - Context

request - PathClientsRequest

Returns:

PathClientsResponse

Throws:

[ApiException](#)

- pathsClients

- public static [PathsClientsResponse](#) pathsClients([ApiContext](#) ctx,
- [PathsClientsRequest](#) request)  
throws [ApiException](#)

Paths Clients  
Get clients up to root

PATH: [{api}/v1/directory/client/pathsClients](#)

TYPE: AUTHENTICATION

PERMISSIONS:

mandatory: []

conditional: [directory.client.entity.read.self, directory.client.entity.read.child, directory.client.products.read]

Parameters:

ctx - Context

request - PathsClientsRequest

Returns:

[PathsClientsResponse](#)

Throws:

[ApiException](#)

## • setClientProducts

- public static [SetClientProductsResponse](#) setClientProducts([ApiContext](#) ctx,
- [SetClientProductsRequest](#) request)  
throws [ApiException](#)

Set Client Products  
Set Client Products

PATH: [{api}/v1/directory/client/setClientProducts](#)

TYPE: AUTHENTICATION

PERMISSIONS:

mandatory: [directory.client.entity.read.child, directory.client.entity.update, directory.client.products.read, directory.client.products.write]

conditional: []

Parameters:

ctx - Context

request - SetClientProductsRequest

Returns:

SetClientProductsResponse

Throws:

[ApiException](#)

- addClientProducts

- public static [AddClientProductsResponse](#) addClientProducts([ApiContext](#) ctx,
- [AddClientProductsRequest](#) request)  
throws [ApiException](#)

Add Client Products

Add Client Products

PATH: [{api}/v1/directory/client/addClientProducts](#)

TYPE: AUTHENTICATION

PERMISSIONS:

mandatory: [directory.client.entity.read.child, directory.client.entity.update, directory.client.products.read, directory.client.products.write]

conditional: []

Parameters:

ctx - Context

request - AddClientProductsRequest

Returns:

AddClientProductsResponse

Throws:

[ApiException](#)

- removeClientProducts

- public static [RemoveClientProductsResponse](#) removeClientProducts([ApiContext](#) ctx,
- [RemoveClientProductsRequest](#) request)  
throws [ApiException](#)

Remove Client Products  
Remove Client Products

PATH: [{api}/v1/directory/client/removeClientProducts](#)

TYPE: AUTHENTICATION

PERMISSIONS:

mandatory: [directory.client.entity.read.child, directory.client.entity.update, directory.client.products.read, directory.client.products.write]

conditional: []

Parameters:

ctx - Context

request - [RemoveClientProductsRequest](#)

Returns:

[RemoveClientProductsResponse](#)

Throws:

[ApiException](#)

- [getClientsProducts](#)

- public static [GetClientsProductsResponse](#) [getClientsProducts](#)([ApiContext](#) ctx,  
[GetClientsProductsRequest](#) request)
- throws [ApiException](#)

Get Clients Products  
Get Clients Products

PATH: [{api}/v1/directory/client/getClientsProducts](#)

TYPE: AUTHENTICATION

PERMISSIONS:

mandatory: [directory.client.products.read]

conditional: [directory.client.entity.read.child, directory.client.entity.read.self]

Parameters:

ctx - Context

request - [GetClientsProductsRequest](#)

Returns:



## GetClientsProductsResponse

Throws:

[ApiException](#)

### • setClientScheme

- public static [SetClientSchemeResponse](#) setClientScheme([ApiContext](#) ctx,  
• [SetClientSchemeRequest](#) request)  
throws [ApiException](#)

Set Client Scheme  
Set Client Scheme

PATH: [{api}/v1/directory/client/setClientScheme](#)

TYPE: AUTHENTICATION

PERMISSIONS:

mandatory: [directory.client.entity.read.child, directory.client.scheme.write]

conditional: [directory.client.products.read]

Parameters:

ctx - Context

request - SetClientSchemeRequest

Returns:

[SetClientSchemeResponse](#)

Throws:

[ApiException](#)

### • addClientContracts

- public static [AddClientContractsResponse](#) addClientContracts([ApiContext](#) ctx,  
• [AddClientContractsRequest](#) request)  
throws [ApiException](#)

Add Client Contracts  
Add Client Contracts

PATH: [{api}/v1/directory/client/addClientContracts](#)

TYPE: AUTHENTICATION

PERMISSIONS:

mandatory: [directory.client.entity.read.child, directory.client.entity.update, directory.client.contracts.read, directory.client.contracts.write]

conditional: []

Parameters:

ctx - Context

request - AddClientContractsRequest

Returns:

[AddClientContractsResponse](#)

Throws:

[ApiException](#)

- **removeClientContracts**

- public static [RemoveClientContractsResponse](#) removeClientContracts([ApiContext](#) ctx,
- [RemoveClientContractsRequest](#) request)  
throws [ApiException](#)

Remove Client Contracts

Remove Client Contracts

PATH: [{api}/v1/directory/client/removeClientContracts](#)

TYPE: AUTHENTICATION

PERMISSIONS:

mandatory: [directory.client.entity.read.child, directory.client.entity.update, directory.client.contracts.read, directory.client.contracts.write]

conditional: []

Parameters:

ctx - Context

request - RemoveClientContractsRequest

Returns:

[RemoveClientContractsResponse](#)

Throws:

[ApiException](#)

- updateClientAddress

- public static [UpdateClientAddressResponse](#) updateClientAddress([ApiContext](#) ctx,  
• [UpdateClientAddressRequest](#) request)  
throws [ApiException](#)

update Client Address  
update Client Address

PATH: [{api}/v1/directory/client/updateClientAddress](#)

TYPE: AUTHENTICATION

PERMISSIONS:

mandatory: [directory.client.entity.read.child, directory.client.entity.update]

conditional: [directory.client.contracts.read, directory.client.products.read]

Parameters:

ctx - Context

request - [UpdateClientAddressRequest](#)

Returns:

[UpdateClientAddressResponse](#)

Throws:

[ApiException](#)

- setClientRisks

- public static [SetClientRisksResponse](#) setClientRisks([ApiContext](#) ctx,  
• [SetClientRisksRequest](#) request)  
throws [ApiException](#)

Set Client Risk Management  
Set Client Risk Management

PATH: [{api}/v1/directory/client/setClientRisks](#)

TYPE: AUTHENTICATION

PERMISSIONS:

mandatory: [directory.client.entity.read.child, directory.client.risks.write, directory.client.risks.read, directory.client.entity.update]  
conditional: []

Parameters:

ctx - Context

request - SetClientRisksRequest

Returns:

SetClientRisksResponse

Throws:

[ApiException](#)

- activateClient

- public static [ActivateClientResponse](#) activateClient([ApiContext](#) ctx,  
• [ActivateClientRequest](#) request)  
throws [ApiException](#)

Activate Client

Activate a Client in the system.

PATH: {api}/v1/directory/client/activateClient

TYPE: AUTHENTICATION

PERMISSIONS:

mandatory: [directory.client.entity.compliance.activate]

conditional: []

Parameters:

ctx - Context

request - ActivateClientRequest

Returns:

ActivateClientResponse

Throws:

[ApiException](#)

- deactivateClient

- public static [DeactivateClientResponse](#) deactivateClient([ApiContext](#) ctx,  
• [DeactivateClientRequest](#) request)  
throws [ApiException](#)

Deactivate Client

Deactivate a Client in the system.

PATH: [{api}/v1/directory/client/deactivateClient](#)

TYPE: AUTHENTICATION

PERMISSIONS:

mandatory: [directory.client.entity.compliance.deactivate]

conditional: []

Parameters:

ctx - Context

request - [DeactivateClientRequest](#)

Returns:

[DeactivateClientResponse](#)

Throws:

[ApiException](#)

## • setClientWebhook

- public static [SetClientWebhookResponse](#) setClientWebhook([ApiContext](#) ctx,  
• [SetClientWebhookRequest](#) request)  
throws [ApiException](#)

Set webhook for a Client

Set webhook for a Client

PATH: [{api}/v1/directory/client/setClientWebhook](#)

TYPE: AUTHENTICATION

PERMISSIONS:

mandatory: [directory.client.entity.read.self, directory.client.webhook.write]

conditional: []

Parameters:

ctx - Context

request - SetClientWebhookRequest

Returns:

SetClientWebhookResponse

Throws:

[ApiException](#)

- renewClientWebhookKey

- public static [RenewClientWebhookKeyResponse](#) renewClientWebhookKey([ApiContext](#) ctx,  
• [RenewClientWebhookKeyRequest](#) request)  
throws [ApiException](#)

Renew webhook key for webhook  
Set webhook for webhook

PATH: {api}/v1/directory/client/renewClientWebhookKey

TYPE: AUTHENTICATION

PERMISSIONS:

mandatory: [directory.client.entity.read.self, directory.client.webhook.key.renew]

conditional: []

Parameters:

ctx - Context

request - RenewClientWebhookKeyRequest

Returns:

RenewClientWebhookKeyResponse

Throws:

[ApiException](#)

- getClientWebhook

- public static [GetClientWebhookResponse](#) getClientWebhook([ApiContext](#) ctx,  
• [GetClientWebhookRequest](#) request)  
throws [ApiException](#)

Get webhook for a client  
Get webhook for a client

PATH: [{api}/v1/directory/client/getClientWebhook](#)

TYPE: AUTHENTICATION

PERMISSIONS:

mandatory: [directory.client.entity.read.self, directory.client.webhook.read]

conditional: []

Parameters:

ctx - Context

request - [GetClientWebhookRequest](#)

Returns:

[GetClientWebhookResponse](#)

Throws:

[ApiException](#)

- deleteClientWebhook

- public static [DeleteClientWebhookResponse](#) deleteClientWebhook([ApiContext](#) ctx,  
• [DeleteClientWebhookRequest](#) request)  
throws [ApiException](#)

Delete webhook for a client  
Delete webhook for a client

PATH: [{api}/v1/directory/client/deleteClientWebhook](#)

TYPE: AUTHENTICATION

PERMISSIONS:

mandatory: [directory.client.entity.read.self, directory.client.webhook.delete]

conditional: []

Parameters:

ctx - Context

request - [DeleteClientWebhookRequest](#)

Returns:

## DeleteClientWebhookResponse

Throws:

[ApiException](#)

- **getChildren**

- public static [GetChildrenResponse](#) getChildren([ApiContext](#) ctx,  
• [GetChildrenRequest](#) request)  
throws [ApiException](#)

Get Children  
Get client children

PATH: [{api}/v1/directory/client/getChildren](#)

TYPE: AUTHENTICATION

PERMISSIONS:

mandatory: [directory.client.entity.read.child]

conditional: [directory.client.products.read, directory.client.contracts.read]

Parameters:

ctx - Context

request - GetChildrenRequest

Returns:

[GetChildrenResponse](#)

Throws:

[ApiException](#)

- **createClient**

- public static [CreateClientResponse](#) createClient([ApiContext](#) ctx,  
• [CreateClientRequest](#) request)  
throws [ApiException](#)

Create Client  
Creates a Client in the system.



PATH: [{api}](#)/v1/directory/client/createClient

TYPE: AUTHENTICATION

PERMISSIONS:

mandatory: [directory.client.entity.read.child, directory.client.entity.create]

conditional: [directory.client.products.read, directory.client.entity.create.bank]

Parameters:

ctx - Context

request - CreateClientRequest

Returns:

CreateClientResponse

Throws:

[ApiException](#)