

Design Resources

This document specifies the API for accessing design data.

- Resources Design
 - Retrieve Shop DesignCategory List
 - Retrieve Shop DesignCategory Designs List
 - Retrieve Shop Designs List
 - Retrieve Shop Design Entity
 - Create Shop Design
 - Update Shop Design
 - Delete Shop Design
 - Retrieve User DesignCategory List
 - Retrieve User DesignCategory Designs List
 - Retrieve User Designs List
 - Retrieve User Design Entity
 - Create User Design
 - Update User Design
 - Delete User Design
- Representations
 - Design
 - DesignCategory
- Use Cases
 - Design Upload (Clean Version)
 - 1.+2. Create Design
 - 3.+4. Get Design Data (Optional)
 - 5. Update Image
 - 6.+7. Get Design Data from API
 - 8. Validate Design
 - 9. Store Design
 - 10.+11. Update Design Meta Data via API
 - 12. Return Status
 - Design Data Update
 - Design Removal

Resources Design

Retrieve Shop DesignCategory List

Uri	.../v1/shops/{shopId}/designCategories
Sample	https://api.spreadshirt.net/api/v1/shops/40000/designCategories
Type	LIST
Method	GET
Special Status Codes	
API Key	yes

Search Request:

Use **?query=...** to search on design categories and filter those categories that contain relevant designs. Filters marketplace design categories only!

The design search supports a fixed set of fields that can be used to further narrow down what you or your customer is looking for. The supported fields and their meaning are as follows:

- **Vector** – Display results for vector designs only. Valid values are *true* or *false*.
An example query would be <https://api.spreadshirt.net/api/v1/shops/205909/designs?query=herz%20mit%20pfeil%20+vector:%28true%29>.
- **Price** – Display results for fixed price or price range. Valid values are all prices, such as *1,59*.
An example queries would be <https://api.spreadshirt.net/api/v1/shops/205909/designs?query=herz%20mit%20pfeil%20+price:%280.00%29> or <https://api.spreadshirt.net/api/v1/shops/205909/designs?query=herz%20mit%20pfeil%20+price:%5B%20TO%203.00%5D>.
- **Date Created** – Display results for designs created between two dates. Valid values are all XML date values, such as *2010-08-16 10:01*.
An example query is <https://api.spreadshirt.net/api/v1/shops/205909/designs?query=herz%20mit%20pfeil%20+created:%5B%20TO%202010-09-01%5D>.
- **Date Modified** – Display results for designs modified between two dates. Valid values are all XML date values, such as *2010-08-16 10:01*.
An example query is <https://api.spreadshirt.net/api/v1/shops/205909/designs?query=herz%20mit%20pfeil%20+modified:%5B%20TO%202010-09-01%5D>.
- **User Ids** – Display results for specific user only. Valid values are all Spreadshirt user ids, such as *1* or *40000*.
An example query would be <https://api.spreadshirt.net/api/v1/shops/205909/designs?query=herz%20mit%20pfeil%20+userIds:%282068606%29>.

- Category Ids** – Display results for specific design category only. Valid values are 1000187, 1000188 or 1000189. An example query would be <https://api.spreadshirt.net/api/v1/shops/205909/designs?query=herz%20mit%20pfeil%20+categoryIds:%281000187%29>.

See LIST type description to see which other query parameters can be used in conjunction with search *query* parameter, such as *sortOrder*, *sortField* or *spellcheck*.

Response Representation:

```
<designCategories ...>
...
</designCategories>
```

See [Design Category Representation](#)

Retrieve Shop DesignCategory Designs List

Url	.../v1/shops/{shopId}/designCategories/{designCategoryId}/designs
Sample	https://api.spreadshirt.net/api/v1/shops/40000/designCategories/501/designs
Type	LIST
Method	GET
Special Status Codes	
API Key	yes

Search Request:

See retrieve shop designs list ...

Response Representation:

```
<designs ...>
...
</designs>
```

See [Design Representation](#)

Retrieve Shop Designs List

Designs are all shop designs or in case of marketplace shop all designs of bestseller category, i.e. in that case /shops/{shopId}/designs is a link to /shops/{shopId}/designCategories/600/designs.

Url	.../v1/shops/{shopId}/designs
Sample	https://api.spreadshirt.net/api/v1/shops/40000/designs
Type	LIST
Method	GET
Special Status Codes	
API Key	yes

Search Request:

Use **?query=...** to search on marketplace designs and filter relevant designs. Works for marketplace shops (eu = 205909) only! See note below: shops /205909/designs is only a link to /shops/205909/designCategories/600/designs!

The design search supports a fixed set of fields that can be used to further narrow down what you or your customer is looking for. The supported fields and their meaning are as follows:

- **Vector** – Display results for vector designs only. Valid values are *true* or *false*.
An example query would be <https://api.spreadshirt.net/api/v1/shops/205909/designs?query=herz%20mit%20pfeil%20+vector:%28true%29>.
- **Price** – Display results for fixed price or price range. Valid values are all prices, such as *1,59*.
An example queries would be <https://api.spreadshirt.net/api/v1/shops/205909/designs?query=herz%20mit%20pfeil%20+price:%280.00%29> or <http://api.spreadshirt.net/api/v1/shops/205909/designs?query=herz%20mit%20pfeil%20+price:%5B%20TO%203.00%5D>.
- **Date Created** – Display results for designs created between two dates. Valid values are all XML date values, such as *2010-08-16 10:01*.
An example query is <https://api.spreadshirt.net/api/v1/shops/205909/designs?query=herz%20mit%20pfeil%20+created:%5B%20TO%202010-09-01%5D>.
- **Date Modified** – Display results for designs modified between two dates. Valid values are all XML date values, such as *2010-08-16 10:01*.
An example query is <https://api.spreadshirt.net/api/v1/shops/205909/designs?query=herz%20mit%20pfeil%20+modified:%5B%20TO%202010-09-01%5D>.
- **User Ids** – Display results for specific user only. Valid values are all Spreadshirt user ids, such as *1* or *40000*.
An example query would be <https://api.spreadshirt.net/api/v1/shops/205909/designs?query=herz%20mit%20pfeil%20+userIds:%282068606%29>.
- **Category Ids** – Display results for specific design category only. Valid values are *1000187*, *1000188* or *1000189*.
An example query would be <https://api.spreadshirt.net/api/v1/shops/205909/designs?query=herz%20mit%20pfeil%20+categoryIds:%281000187%29>.

See LIST type description to see which other query parameters can be used in conjunction with search *query* parameter, such as *sortOrder*, *sortField* or *spellcheck*.

Response Representation:

```
<designs ...>
...
</designs>
```

See [Design Representation](#)

Retrieve Shop Design Entity

Url	.../v1/shops/{shopId}/designs/{designId}
Sample	https://api.spreadshirt.net/api/v1/shops/40000/designs/m4298936-1
Type	ENTITY
Method	GET
Special Status Codes	
API Key	yes

Response Representation:

```
<design ...>
...
</design>
```

See [Design Representation](#)

Create Shop Design

Create shop design. Initial state is without uploaded design image. However, design image can be uploaded or modified. A design commission (design /price) cannot be set. This is only possible for user design.

Url	.../api/v1/shops/{shopId}/designs
Sample	https://api.spreadshirt.net/api/v1/shops/40000/designs
Type	LIST
Method	POST
Special Status Codes	
APIKey	yes

SessionId	no (yes in case user is logged in)
------------------	------------------------------------

Request Representation:

```
<design ...>
  ...
</design>
```

See [Design Representation](#)

Response Representation:

Header:

```
Location: ...api/v1/shops/42/designs/789
```

Update Shop Design

Url	.../api/v1/shops/{shopId}/designs/{designId}
Sample	https://api.spreadshirt.net/api/v1/shops/40000/designs/10815
Type	ENTITY
Method	PUT
Special Status Codes	
APIKey	yes
SessionId	no (yes in case user is logged in)

Request Representation:

```
<design ...>
  ...
</design>
```

See [Design Representation](#)

Delete Shop Design

Url	.../api/v1/shops/{shopId}/designs/{designId}
Sample	https://api.spreadshirt.net/api/v1/shops/40000/designs/10815
Type	ENTITY
Method	DELETE
Special Status Codes	
APIKey	yes
SessionId	no (yes in case user is logged in)

Retrieve User DesignCategory List

Url	.../v1/users/{userId}/designCategories
Sample	https://api.spreadshirt.net/api/v1/users/40000/designCategories

Type	LIST
Method	GET
Special Status Codes	
APIKey	yes
SessionId	yes

Response Representation:

```
<designCategories ...>
  ...
</designCategories>
```

See [Design Category Representation](#)

Retrieve User DesignCategory Designs List

Url	.../v1/users/{userId}/designCategories/{designCategoryId}/designs
Sample	https://api.spreadshirt.net/api/v1/users/40000/designCategories/501/designs
Type	LIST
Method	GET
Special Status Codes	
APIKey	yes
SessionId	yes

Response Representation:

```
<designs ...>
  ...
</designs>
```

See [Design Representation](#)

Retrieve User Designs List

Returns all grabbed and own designs of a user.

Url	.../v1/users/{userId}/designs
Sample	https://api.spreadshirt.net/api/v1/users/40000/designs
Type	LIST
Method	GET
Special Status Codes	
APIKey	yes
SessionId	yes

Response Representation:

```
<designs ...>
...
</designs>
```

See [Design Representation](#)

Retrieve User Design Entity

Url	.../v1/users/{userId}/designs/{designId}
Sample	https://api.spreadshirt.net/api/v1/users/400000/designs/6664977
Type	ENTITY
Method	GET
Special Status Codes	
APIKey	yes
SessionId	yes

Response Representation:

```
<design ...>
...
</design>
```

See [Design Representation](#)

Create User Design

Create user design. Use an active session in order to make your uploaded designs visible in your Spreadshirt user area under Designs > Show all. This creates an "empty" design and responds with its design ID. In the next step the actual image (pixel / vector file) needs to be uploaded using this design ID => <https://developer.spreadshirt.net/display/API/Image+Resources#ImageResources-Update%28Upload%29DesignImage> Later one can also modify the design by its ID. A design commission (design/price) can be set for user designs.

Url	.../api/v1/users/{userId}/designs
Sample	https://api.spreadshirt.net/api/v1/users/40000/designs
Type	LIST
Method	POST
Special Status Codes	
APIKey	yes
SessionId	yes

Request Representation:

```
<design ...>
...
</design>
```

See [Design Representation](#)

Response Representation:

Header:

```
Location: ...api/v1/users/42/designs/789
```

Update User Design

Url	.../api/v1/users/{userId}/designs/{designId}
Sample	https://api.spreadshirt.net/api/v1/users/40000/designs/10815
Type	ENTITY
Method	PUT
Special Status Codes	
APIKey	yes
SessionId	yes

Request Representation:

```
<design>
  ...
</design>
```

See [Design Representation](#)

Delete User Design

Url	.../api/v1/users/{userId}/designs/{designId}
Sample	https://api.spreadshirt.net/api/v1/users/40000/designs/10815
Type	ENTITY
Method	DELETE
Special Status Codes	
APIKey	yes
SessionId	yes

Representations

Design

Field	Descripton
design	Represents a platform design.
design /@weight	The weight of the design in a list. Used for sorting designs in a list.
design /name	The i18n name of the design in case of a marketplace design or the normal design name in case of shop/user designs.
design /description	The i18n description of the design in case of a marketplace design or the normal design description in case of shop/user designs.
design /sourceUrl	Reference to the actual source of the image to be uploaded. Can be used together with uploading design references at the image server (pointer to images that will be used to fetch image automatically) instead of uploading the actual image to avoid duplicate uploads of the same image. Return code is in this case 201 if no usable design with sourceUrl exists or 200 if usable design with same sourceUrl exists already. Location header contains the URL to the actual design in both cases.

design /tags	The i18n tags of the design in case of a marketplace design or the normal design tags in case of shop/user designs.
design /user	Reference to the user that owns the design.
design /restrictions	The restrictions tell you what you can't do with the design.
design /restrictions /fixedColors	FixedColors means you can't change the colors of a vector design.
design /restrictions /colorCount	Number of colors for the different color layers of a vector design.
design /restrictions /movable	Movable means you can move the design.
design /restrictions /minimumScale	MinimumScale is the scale max scale by which you can downsize the design image.
design /restrictions /denyOtherText	Defines whether this design can be used together with other text configuration on print area or product. Valid values are <i>no</i> , <i>onProduct</i> , <i>onPrintArea</i> .
design /restrictions /denyOtherDesigns	Defines whether this design can be used together with other design configuration on print area or product. Valid values are <i>no</i> , <i>onProduct</i> , <i>onPrintArea</i> .
design /restrictions /allowScale	Defines whether this design can be scaled (svg scale).
design /restrictions /allowFlip	Defines whether this design can be flipped (svg scale x + y).
design /restrictions /allowRotate	Defines whether this design can be rotated (svg rotate).
design /size	Upload size of the image in mm.
design /colors	List of colors for the different color layers of a vector design, i.e. svg, cdr, ai.
design /colors /color	Configuration for one color of a vector design.
design /colors /color /default	The default color of the color layer used when displaying that design in a shop.
design /colors /color /origin	The actual color of that color layer on image upload.

design /colors /color /resources	List of connected resources (aka images). Use the links in case you need to display an image for the color layer.
design /colors /color /resources /resource	
design /printTypes	List of references to print types with which the design can be printed.
design /printTypes /printType	Reference to a print type, e.g. flex, flock, digi, with which the design can be printed. See Retrieve Shop Print Type for more details.
design /designCategories	List of design categories to which the design is assigned to (either shop or marketplace categories).
design /designCategories /designCategory	
design /price	The design price in the currency of the shop. This is actually the design commission that the owner of the design will earn once an article holding this design is sold. Can be set for user designs only.
design /resources	List of connected resources (aka images). Use the links in case you need to display an image for the design. Please note that we can display vector designs as pixel designs only for legal reasons (means we can't deliver an svg there).
design /resources /resource	
design /created	date when design was created
design /modified	date when design was last modified

Example:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<design xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns="http://api.spreadshirt.net" weight="0.0"
  xlink:href="https://api.spreadshirt.net/api/v1/shops/205909/designs/m10004349-2" id="m10004349-2">
  <name>vikings</name>
  <description>vikings</description>
  <tags>vikings</tags>
  <user xlink:href="https://api.spreadshirt.net/api/v1/users/2343121" id="2343121"/>
  <restrictions>
    <fixedColors>>false</fixedColors>
    <ownText>>true</ownText>
    <colorCount>0</colorCount>
    <obligatory>>false</obligatory>
    <movable>>true</movable>
    <targetView>0</targetView>
    <minimumScale>70</minimumScale>
    <visible>>true</visible>
    <inverted>>false</inverted>
    <denyOtherText>no</denyOtherText>
    <denyOtherDesigns>no</denyOtherDesigns>
    <allowScale>>false</allowScale>
    <allowFlip>>false</allowFlip>
    <allowRotate>>false</allowRotate>  </restrictions>
  <size unit="mm">
    <width>200.871666666667</width>
    <height>184.15</height>
  </size>
</design>
```

```

<colors>
  <color>
    <default>#000000</default>
    <origin>#000000</origin>
    <resources>
      <resource xlink:href="https://image.spreadshirtmedia.net/image-server/v1/designs/10004349,colors
[1]=none,colors[2]=none" type="montage" mediaType="png"/>
    </resources>
  </color>
  ...
</colors>
<printTypes>
  <printType xlink:href="https://api.spreadshirt.net/api/v1/shops/205909/printTypes/19" id="19"/>
  ...
</printTypes>
<designCategories>
  <designCategory id="1000186"/>
  ...
</designCategories>
<price>
  <vatExcluded>3.00</vatExcluded>
  <vatIncluded>3.00</vatIncluded>
  <vat>0</vat>
  <currency xlink:href="https://api.spreadshirt.net/api/v1/currencies/1" id="1"/>
</price>
<resources>
  <resource xlink:href="https://image.spreadshirtmedia.net/image-server/v1/designs/10004349,width=42,
height=42" type="preview" mediaType="png"/>
  <resource xlink:href="https://image.spreadshirtmedia.net/image-server/v1/designs/10004349" type="
montage" mediaType="png"/>
</resources>
  <created>2010-06-07T19:15:11Z</created>
  <modified>2010-07-30T15:17:15Z</modified>
</design>

```

DesignCategory

Field	Description
designCategory	Represents a platform design category.
designCategory/@type	Represents the type of the design category. In shop context, you will find the following types: <ul style="list-style-type: none"> • MY_DESIGNS • UPLOADED - Uploaded designs for this shop. Needs to be configured for this shop (community mode). • MARKETPLACE - Type for marketplace categories. Marketplace categories are categories from our special marketplace shop (eu - 205909, na - 117728) • SHOP - Type for generic shop categories defined by Spreadshirt. • SHOP_CUSTOM - Type of custom shop specific categories created by Spreadshirt partner. • BESTSELLER - Type for a magic category that allows to retrieve an ordered list of all items of the parent's sub categories. In user context, the types will be: <ul style="list-style-type: none"> • MY_DESIGNS • OWN - Designs owned by the partner. • GRABBED - Designs grabbed by the partner.
designCategory/@weight	The weight of the category is used for sorting cartegories in the category tree.
designCategory/name	The i18n name of the category.
designCategory/designs	Reference to the list of designs linked to this category.
designCategory /designCategories	List of sub categories of this design category.

Example:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<designCategory xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns="http://api.spreadshirt.net"

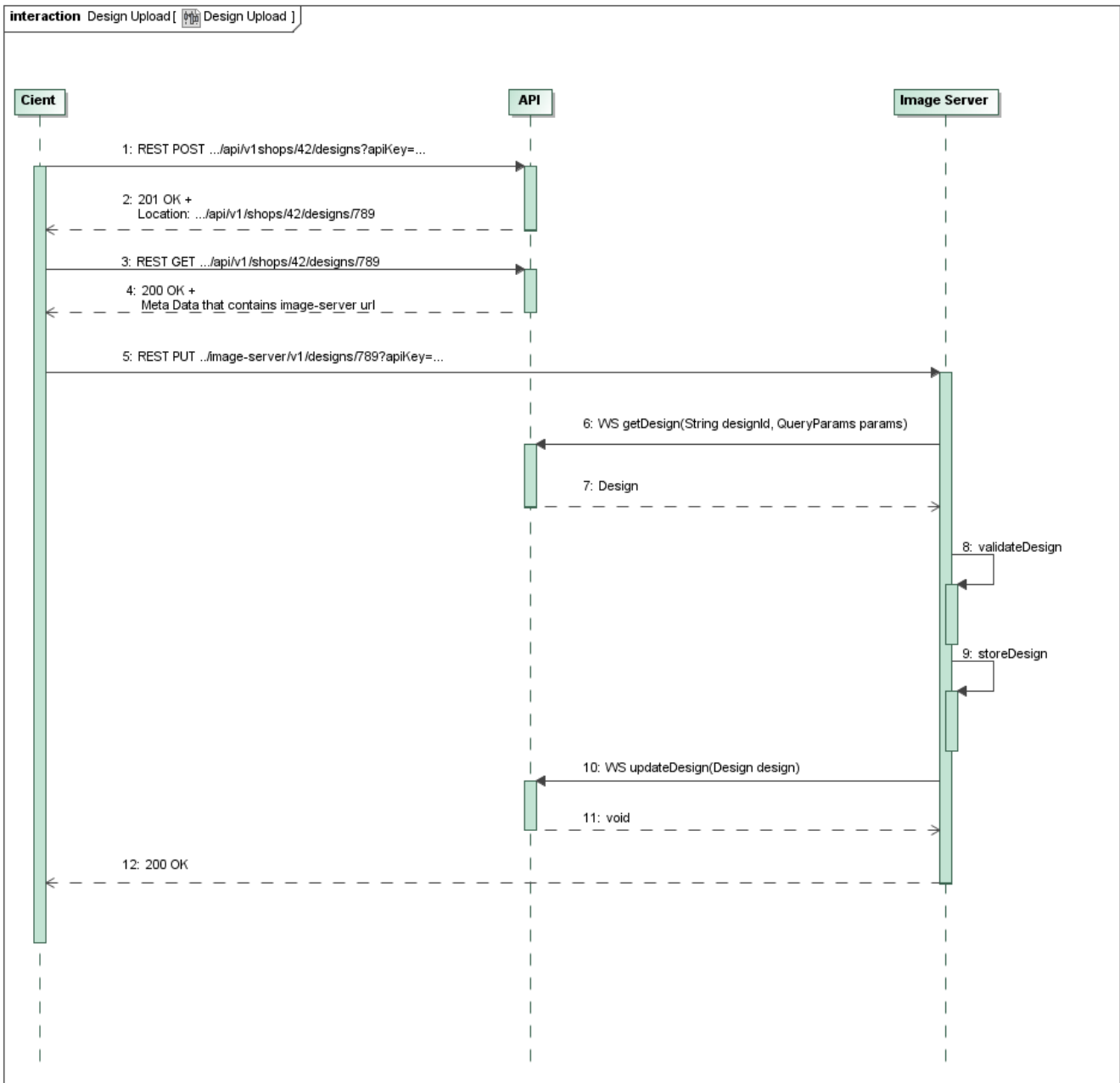
```

```
        type="MARKETPLACE" weight="3.0" id="1000000">
<name>Marketplace Designs</name>
<designs xlink:href="https://api.spreadshirt.net/api/v1/shops/205909/designCategories/1000000/designs"/>
<designCategories>
  <designCategory type="BESTSELLER" weight="-1.0" id="b1000000">
    <name>Bestseller</name>
    <designs xlink:href="https://api.spreadshirt.net/api/v1/shops/205909/designCategories/b1000000
/designs"/>
  </designCategory>
  ...
</designCategories>
...
</designCategory>
```

Use Cases

Design Upload (Clean Version)

Upload design image means you first create the design meta data and then upload the image for that design (which you can do one or more times - imagine broken upload, invalid eps files etc.).



1.+2. Create Design

Create design meta data via API. This creates an "empty" design and responds with its design ID. In the next step the actual image (pixel / vector file) can be uploaded using this design ID and later modified too.

Get **Location** header to the created design meta data.

```
Location: https://api.spreadshirt.net/api/v1/shops/42/designs/789
```

3.+4. Get Design Data (Optional)

Get design data + created image server image upload url.

Contains **URL to image server resource**.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<design xmlns:xlink="http://www.w3.org/1999/xlink" xmlns="http://api.spreadshirt.net" id="789">
  <name>Jumpstyle DE</name>
  <description>Jumpstyle </description>
  ...
  <resources>
    <resource xlink:href="https://image.spreadshirtmedia.net/image-server/v1/designs/789" type="montage"
mediaType="image/png" />
    ...
  </resources>
</design>
```

5. Update Image

Upload image data (eps, svg, gif, jpg, png).

Provide security information, such as apiKey, sig, time (see SprdAuth protocol).

SessionId is required for user designs only.

6.+7. Get Design Data from API

Returns Design meta data. Contains required design id and user id.

8. Validate Design

Image Server receives image data and validates what it got. If uploaded image does not fit, an error message is returned to the user.

9. Store Design

Store design in file structure in user folders using design id and user id. (do we have a better idea here? can we change that? do we want to change that?)

10.+11. Update Design Meta Data via API

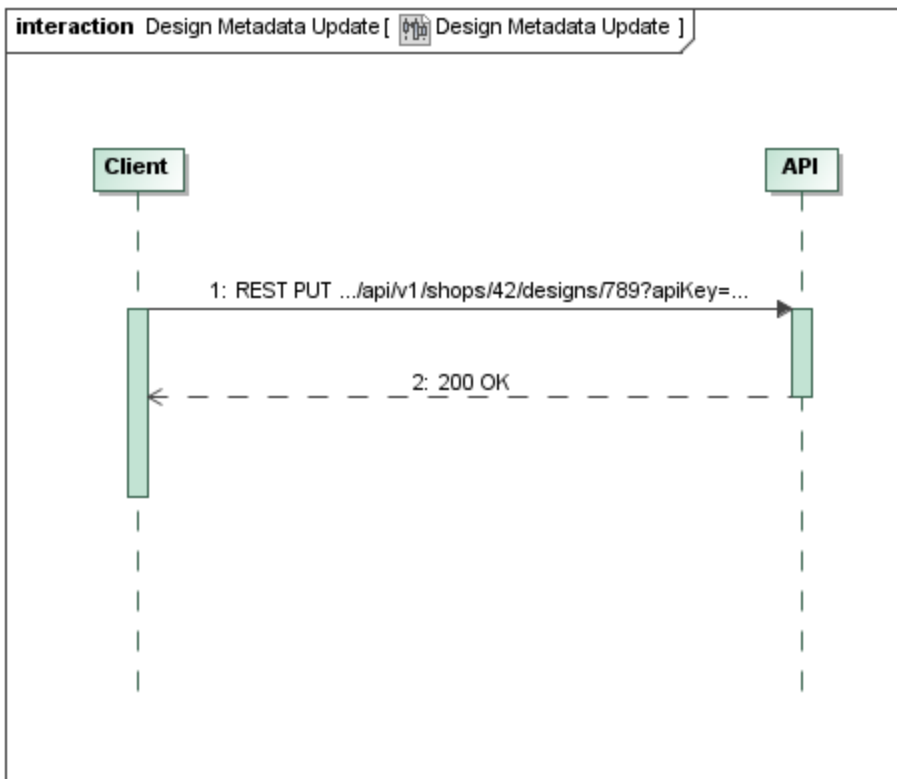
Update Design data, e.g. design file name, file ending etc..

12. Return Status

200 if uploaded image was stored successfully.

Design Data Update

Update existing Design meta data.



Design Removal

Remove Design meta data. Corresponding image will be removed asynchronously.

