

Revision	Date	Author
v0.1	23.10.17	Jan Speckamp
v0.2	05.11.17	Jan Speckamp
v0.3	14.11.17	Jan Speckamp
V0.4	16.01.18	Jens Seifert Jan Speckamp
V0.5	20.01.18	Jan Speckamp
V1.0	28.01.18	Jens Seifert Jan Speckamp

Green denotes fully implemented functional requirements.
Red denotes not implemented functional requirements.

Function	Description										
Data Discovery											
F00010 (LF00020,LF00020, LF00030,LF00040)	<p>The App has a Sidebar Pane titled “Search”. At the top of the Pane several Elements allow creation of Search Criteria:</p> <ul style="list-style-type: none"> • Textinput for dataset name • Label with bbox + Button for creation of bbox • Date Input for earliest recording date • Date Input for latest recording date <p>A Button “Search” is present below the Search criteria. On clicking the Button the client (asynchronously) requests Data from the API via the “/search” Endpoint.</p> <p>The API provides an Endpoint ‘/search’ to GET-Requests with 5 Parameters. The Parameters are ‘substring’, ‘bbox’, ‘startdate’, ‘enddate’, ‘page’.</p> <table border="1"> <tbody> <tr> <td>substring</td> <td>String which has to be included in dataset name</td> </tr> <tr> <td>bbox</td> <td>structure: ‘<minx>, <miny>, <maxx>, <maxy>’ defining box datasets have to intersect with</td> </tr> <tr> <td>startdate</td> <td>Defines earliest date of dataset creation. format: ‘YYYY-MM-DDTHH:MM:SSZ’</td> </tr> <tr> <td>enddate</td> <td>Defines latest date for dataset creation. format: ‘YYYY-MM-DDTHH:MM:SSZ’</td> </tr> <tr> <td>page</td> <td>Defines which page is returned. Pages contain 8 Elements each.</td> </tr> </tbody> </table> <p>The Parameters are appended to the URL: “.../search?substring=<substring>&bbox=<bbox>&startdate=<startdate>&enddate=<enddate>&page=<page>”</p> <p>API returns a JSON Object containing two Arrays labeled “L1C” and “L2A”. These Arrays contain the first 8 (combined count) Objects with all their metadata.</p>	substring	String which has to be included in dataset name	bbox	structure: ‘<minx>, <miny>, <maxx>, <maxy>’ defining box datasets have to intersect with	startdate	Defines earliest date of dataset creation. format: ‘YYYY-MM-DDTHH:MM:SSZ’	enddate	Defines latest date for dataset creation. format: ‘YYYY-MM-DDTHH:MM:SSZ’	page	Defines which page is returned. Pages contain 8 Elements each.
substring	String which has to be included in dataset name										
bbox	structure: ‘<minx>, <miny>, <maxx>, <maxy>’ defining box datasets have to intersect with										
startdate	Defines earliest date of dataset creation. format: ‘YYYY-MM-DDTHH:MM:SSZ’										
enddate	Defines latest date for dataset creation. format: ‘YYYY-MM-DDTHH:MM:SSZ’										
page	Defines which page is returned. Pages contain 8 Elements each.										
F00020 (LF00050)	The results from F00010 are added to the Results Pane of the Sidebar.										

	Red opaque polygons are created on the map based on the bbox of the results showing the spatial extent of the individual datasets.
F00030 (LF00060)	Clicking on the accordion item from F00020 expands the item. Metadata is listed in the expanded item. A 'Visualize'-Button is created at the bottom of the expanded panel. When clicking on a bbox on the map the sidebar is opened and the appropriate accordion item is expanded.

Data visualization

F01010 (LF01010)	A Control Element in the top-right corner of the map allows switching between Base maps. Basemaps are preconfigured and can not be modified.																
F01020 (LF01020)	<p>A Control Element in the bottom-right corner of the map allows zooming. Alternatively zooming using the Scrollwheel is supported.</p> <p>The inbuilt Panning functionality of the map library (Leaflet) is used.</p> <p>The Sentinel imagery is added as a TMS.</p> <p>The API provides an Endpoint for creating the Tiles via a POST-Request to:</p> <pre>'http://{s}.<apiserver>/generate?{<options>}'</pre> <p>with s being set by the mapping library. With <apiserver> being the URL of the APIServer. with the Request Body being a html Form with the following fields:</p> <p>In case of RGB:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">KEY</th> <th style="text-align: center;">VALUE/VALUETYPE</th> </tr> </thead> <tbody> <tr> <td>rgbbool (is rgb requested?)</td> <td>true</td> </tr> <tr> <td>l2a (is level 2A Dataset?)</td> <td>false / true</td> </tr> <tr> <td>tci (is true color image?)</td> <td>false /true</td> </tr> <tr> <td>rcdn (subdataset_name for redchannel)</td> <td><datasetname></td> </tr> <tr> <td>gcdn (subdataset_name for greenchannel)</td> <td><datasetname></td> </tr> <tr> <td>bcdn (subdataset_name for bluechannel)</td> <td><datasetname></td> </tr> <tr> <td>rcn (redchannel_name)</td> <td><bandname></td> </tr> </tbody> </table>	KEY	VALUE/VALUETYPE	rgbbool (is rgb requested?)	true	l2a (is level 2A Dataset?)	false / true	tci (is true color image?)	false /true	rcdn (subdataset_name for redchannel)	<datasetname>	gcdn (subdataset_name for greenchannel)	<datasetname>	bcdn (subdataset_name for bluechannel)	<datasetname>	rcn (redchannel_name)	<bandname>
KEY	VALUE/VALUETYPE																
rgbbool (is rgb requested?)	true																
l2a (is level 2A Dataset?)	false / true																
tci (is true color image?)	false /true																
rcdn (subdataset_name for redchannel)	<datasetname>																
gcdn (subdataset_name for greenchannel)	<datasetname>																
bcdn (subdataset_name for bluechannel)	<datasetname>																
rcn (redchannel_name)	<bandname>																

gcn (greenchannel_name)	<bandname>
bcn (bluechannel_name)	<bandname>
rcmin (redchannel_minvalue)	0-65536
gcmin (greenchannel_minvalue)	0-65536
bcmin (bluechannel_minvalue)	0-65536
rcmax (redchannel_maxvalue)	0-65536
gcmax (greenchannel_maxvalue)	0-65536
bcmax (bluechannel_maxvalue)	0-65536

In case of Greyscale Database:

KEY	VALUE/VALUETYPE
rgbbool (is rgb requested?)	false
l2a (is level 2A Dataset?)	false / true
tci (is true color image?)	false /true
gscdn (subdataset_name for greychannel)	<datasetname>
gsc (greychannel_name)	<bandname>
greymin (greyscale_minvalue)	0-65536
greymax (greyscale_maxvalue)	0-65536

The POST-Request returns a unique id in plain text.

The TMS is included via the custom URL:

'http://<apiserver>/api/data/<id>/{z}/{x}/{-y}.png'

with z, x, y, being set by the mapping library.

With <apiserver> being the URL of the APIServer.

with <id> being the id returned by the Server previously.

F01030
(LF01030)

Radio Buttons above to the 'Visualize'-Button in the results pane allows selecting between visualizing a Greyscale or RGB Image. Upon switching the appropriate HTML-Form Elements described in **F01040** are activated.

F01040
(LF01040)

Next to the 'Visualize'-Button in the Accordion the available color channels options (R+G+B or Grey) are listed with associated drop-down-menus to specify the spectral Bands that should be visualized in that channel.

These Form-Elements correspond to the ones described in **F01020**

F01050 (LF01050)	<p>Next to the dropdown-menus specified in F01040 two Input fields for min and max values are defined.</p> <p>For RGB Images upon entering a value in the Fields the Form Fields 'redchannel_minvalue', 'greenchannel_minvalue', 'bluechannel_minvalue', 'redchannel_maxvalue', 'greenchannel_maxvalue', 'bluechannel_maxvalue' are changed accordingly.</p> <p>For Greyscale Images upon entering a value in the Fields the Form Fields 'greychannel_maxvalue', 'greychannel_maxvalue' are changed.</p> <p>The API maps the given range ('channel_minvalue' - 'channel_maxvalue') in the source data to 0-255.</p>										
F01060 (LF01060)	<p>A slider next to the 'Visualize'-Button in the Accordion Panel adjusts the opacity. The mapping library provides the implementation.</p>										
F01070 (LF01070)	<p>A new layer is created for every visualization with new parameters. (Old layers are discarded). As tile data is streamed by the mapping library no reload is necessary.</p>										
F01080 (LF01080)	<p>The API provides an Endpoint '/value' for GET-Requests with 4 Parameters</p> <table border="1" data-bbox="424 1169 1436 1431"> <thead> <tr> <th>KEY</th> <th>VALUE/VALUETYPE</th> </tr> </thead> <tbody> <tr> <td>d (datasetname)</td> <td><datasetname></td> </tr> <tr> <td>b (bandname)</td> <td><bandname></td> </tr> <tr> <td>x (WGS84 x coordinate)</td> <td>-180 - 180</td> </tr> <tr> <td>y (WGS84 y coordinate)</td> <td>-90 - 90</td> </tr> </tbody> </table> <p>The GET-Request returns the value in plain text. If a TCI Band is requested the values are returned as json array ordered ["redvalue", "greenvalue", "bluevalue"].</p>	KEY	VALUE/VALUETYPE	d (datasetname)	<datasetname>	b (bandname)	<bandname>	x (WGS84 x coordinate)	-180 - 180	y (WGS84 y coordinate)	-90 - 90
KEY	VALUE/VALUETYPE										
d (datasetname)	<datasetname>										
b (bandname)	<bandname>										
x (WGS84 x coordinate)	-180 - 180										
y (WGS84 y coordinate)	-90 - 90										
Miscellaneous											
F02010 (LF02010)	<p>The different functionalities are all available via the single sidebar.</p>										
F02020 (LF02020)	<p>The Sidebar provides a 'Save' Button. Upon clicking the Button a state is created representing the current State of the Application. The string is appended to the base-URL to load the data.</p> <p>Upon loading the Page with a URL containing a state the clientside Javascript interprets this embedded state and loads it.</p>										

The State Object has the following fields:

Name	Value
st (searchterm)	Searchterm currently in the Search Input
sbox (searchbox)	Bbox currently in the Search Input
ssd (searchstartdate)	Startdate currently in the Search Input
sed (searchenddate)	Enddate currently in the Search Input
p (page)	Page (of the Dataset List) currently opened
ser (search_done)	Has search been done
ds (datasets) [Object Array]	
o (opacity)	Opacity
vis (visualized)	True if currently visualized
exp (expanded)	True if currently expanded
btn	RGB or Greyscale selected
gscdn (greychannel_dataset_name)	Name of Subdataset currently assigned to Greyscale
rcdn (redchannel_dataset_name)	Name of Subdataset currently assigned to Red Channel
gcdn (greenchannel_dataset_name)	Name of Subdataset currently assigned to Green Channel
bcdn (bluechannel_dataset_name)	Name of Subdataset currently assigned to Blue Channel
greymax (greyscale_minvalue)	Minimum Value in Grey Channel
rcmin (redchannel_minvalue)	Minimum Value in Red Channel
gcmn (greenchannel_minvalue)	Minimum Value in Green Channel
bcmin (bluechannel_minvalue)	Minimum Value in Blue Channel
greymax (greyscale_maxvalue)	Maximum Value in Grey Channel
rcmax (redchannel_maxvalue)	Maximum Value in Red Channel
gcmax (greenchannel_maxvalue)	Maximum Value in Green Channel
bcmax (bluechannel_maxvalue)	Maximum Value in Blue Channel

Data Processing

F10010
(LX01010,
LX01020)

The sidebar provides a “Processing”-Pane. At the Top a Dropdown Menu allows selection of the Dataset. When Dataset is selected every Band is assigned a Variable (x1 to xN in alphabetical order for default bands, c1 to cN for calculated Bands). They can be used in calculations.

	<p>An input group allows input of mathematical terms and a band name. A button allows creation of additional input groups.</p> <p>A 'add to Bands'-Button adds the Band resulting from the calculation to the Bands listed in the drop-down list described in F01040. Onlick the Sidepanel switches to the Dataset list and expands the affected dataset. The Band is then visualized using F01020.</p> <p>The computation is done on the server-side.</p>
<p>F10030 (LX01030)</p>	<p>TBD</p>
<p>F10040 (LX01040)</p>	<p>TBD</p>