
The Alpine Expedition
PIMCORE INSPIRE

Unveiling the Depths of Pimcore Studio

Topics

Design and UX

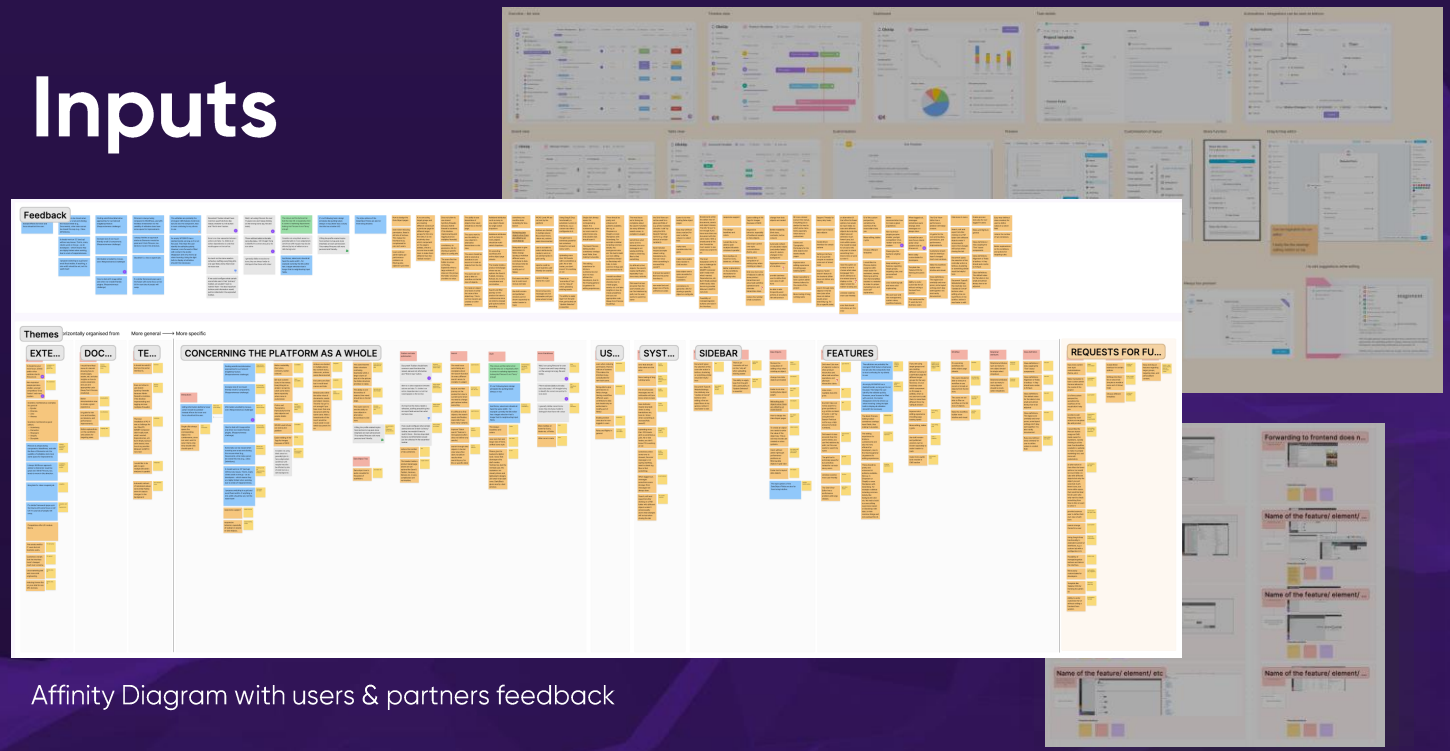
Generic Data Index

Studio API

Studio UI

Design and UX

Inputs

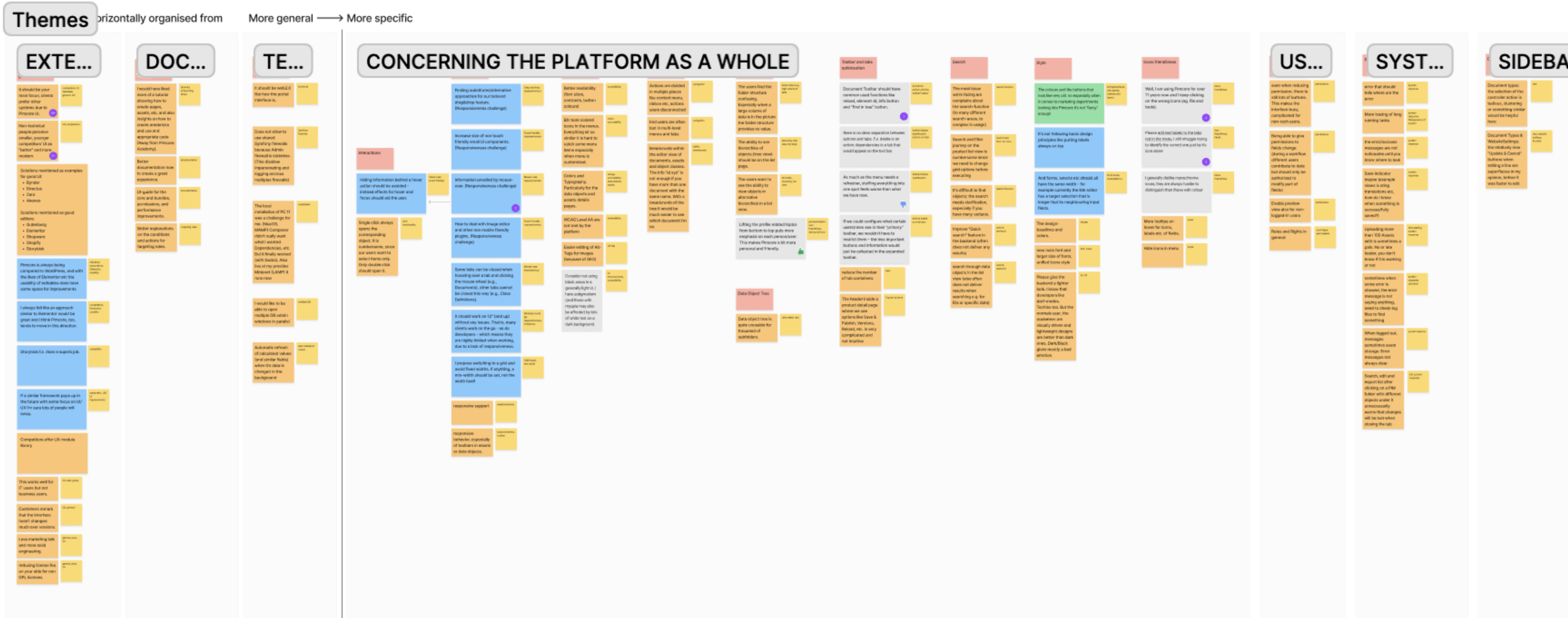


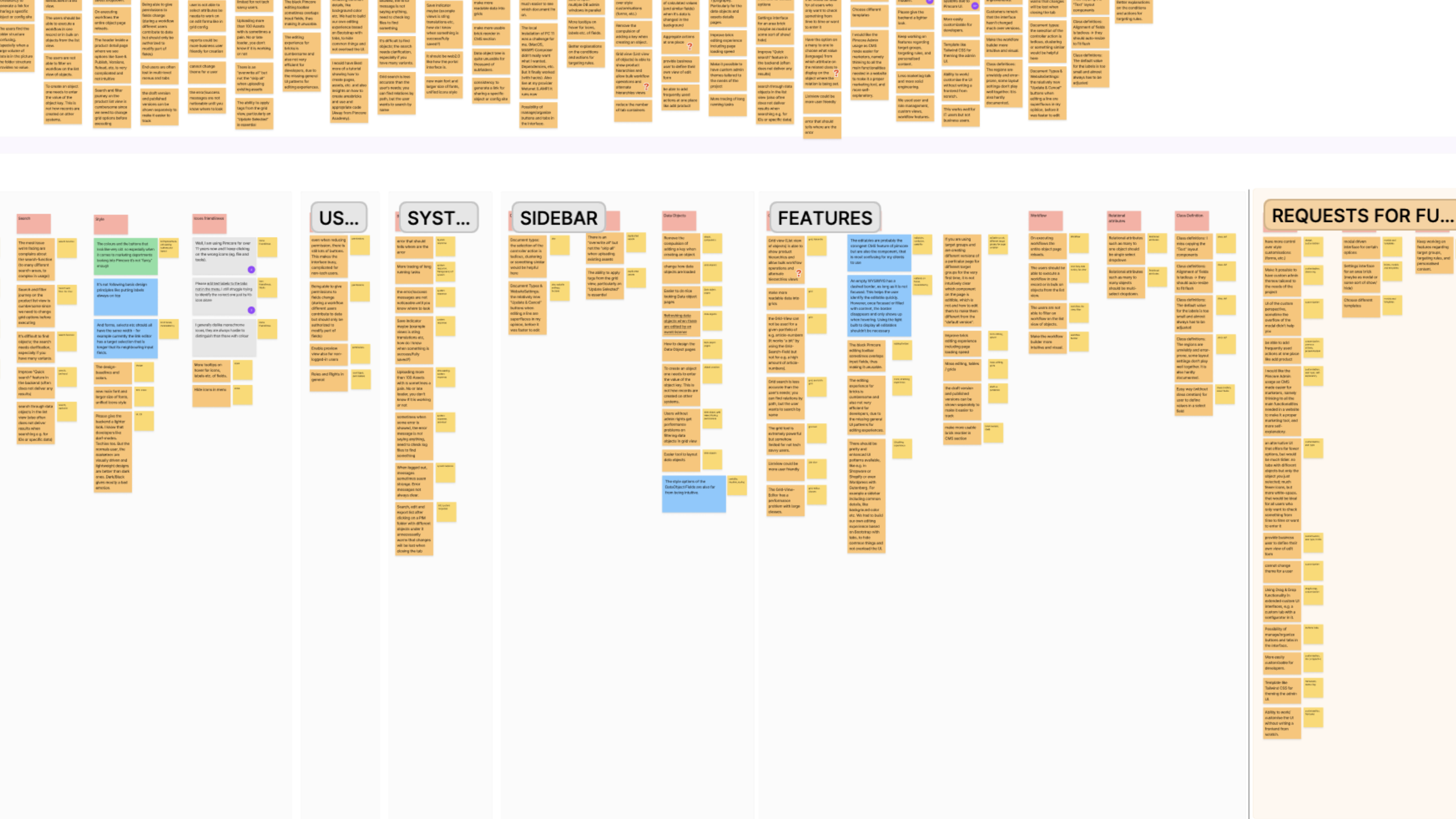
Affinity Diagram with users & partners feedback

- RESEARCH
- SURVEYS
- EXPERIENCE & FEEDBACK
- ACCESSIBILITY

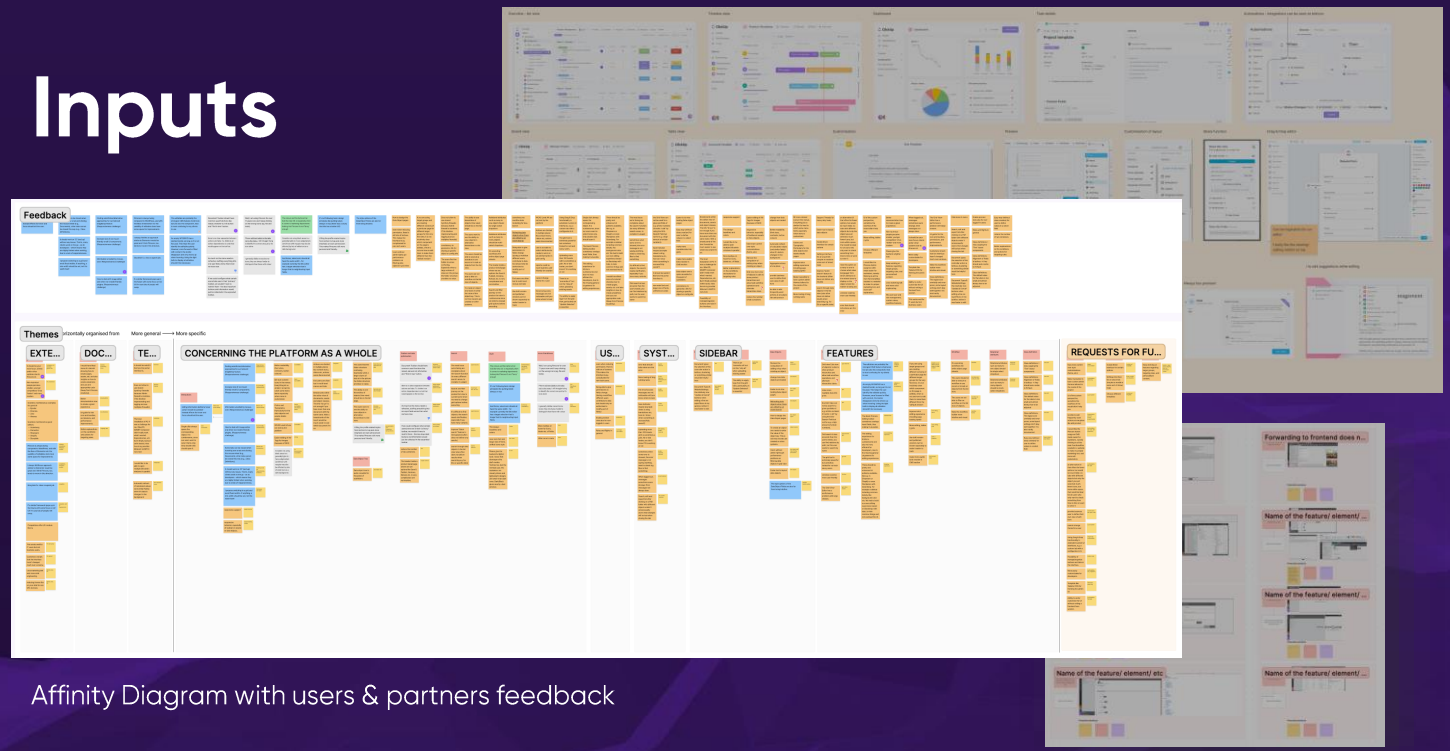
Feedback

Themes




US...
SYST...
SIDEBAR
FEATURES
REQUESTS FOR FU...

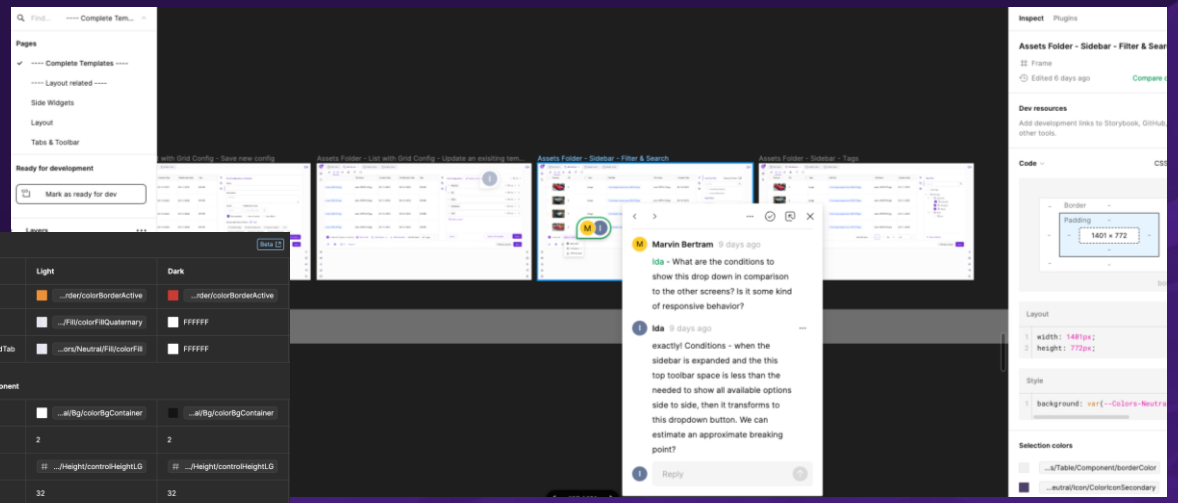
Inputs

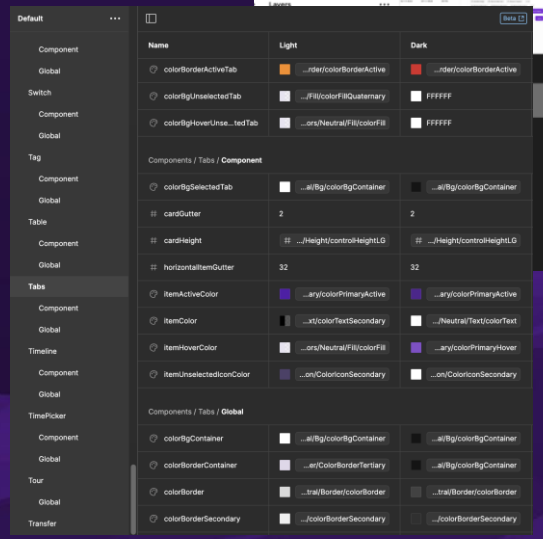


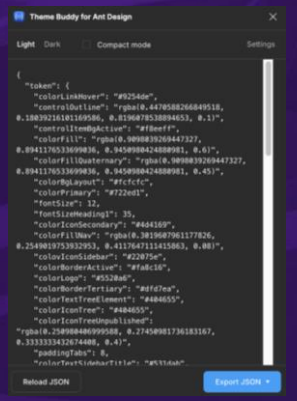
Affinity Diagram with users & partners feedback

- RESEARCH
- SURVEYS
- EXPERIENCE & FEEDBACK
- ACCESSIBILITY

Synergy of tools







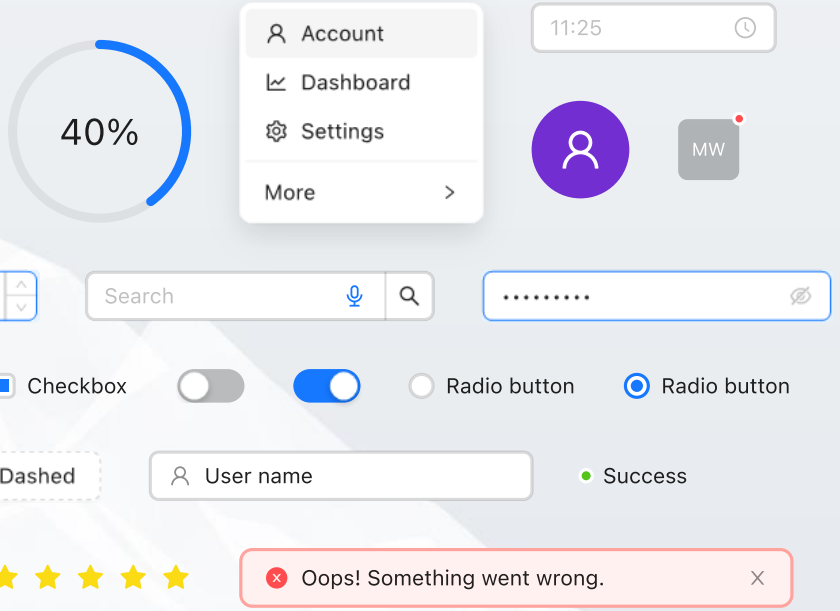
Ant Design

CONSISTENCY

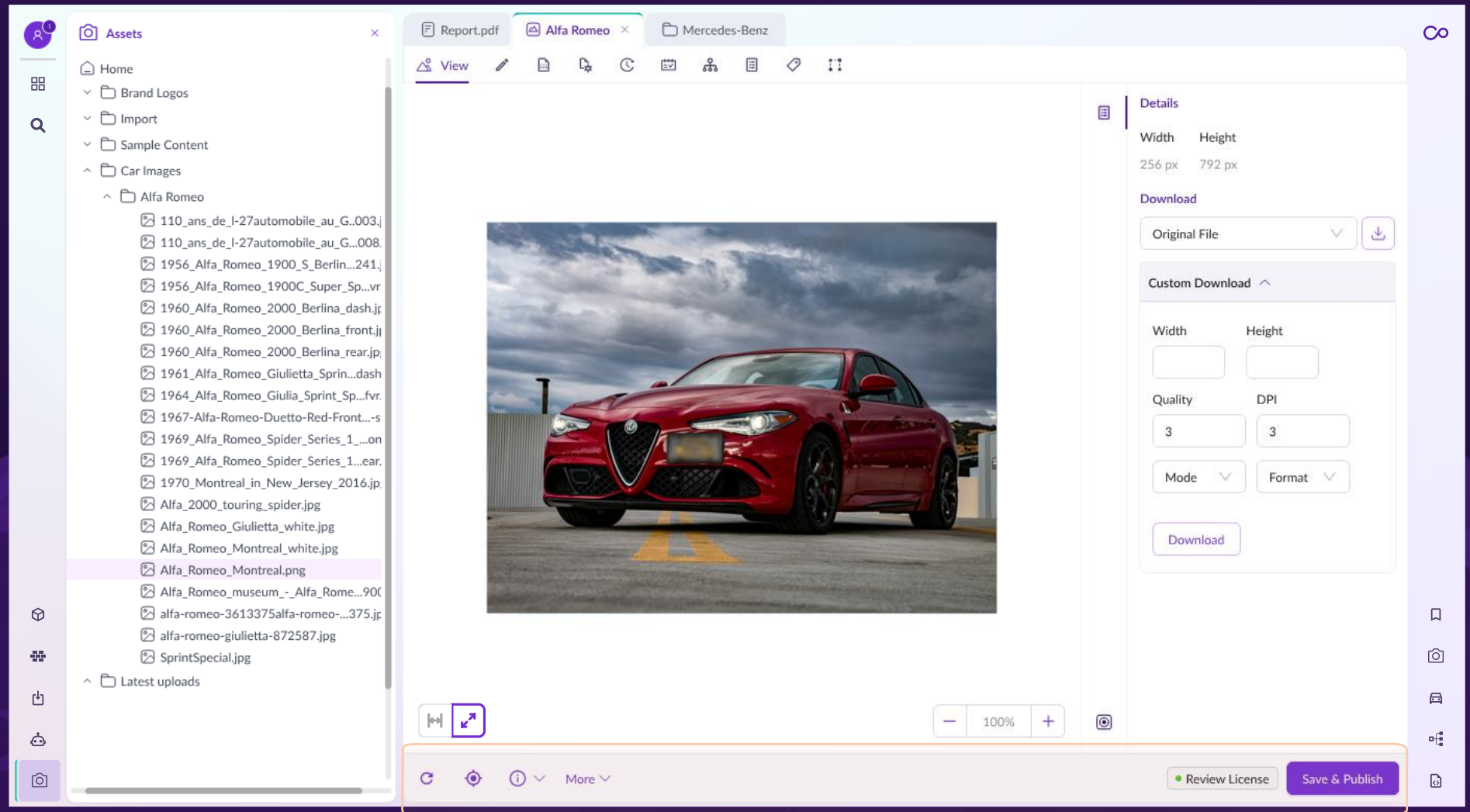
EFFICIENCY

ACCESSIBILITY

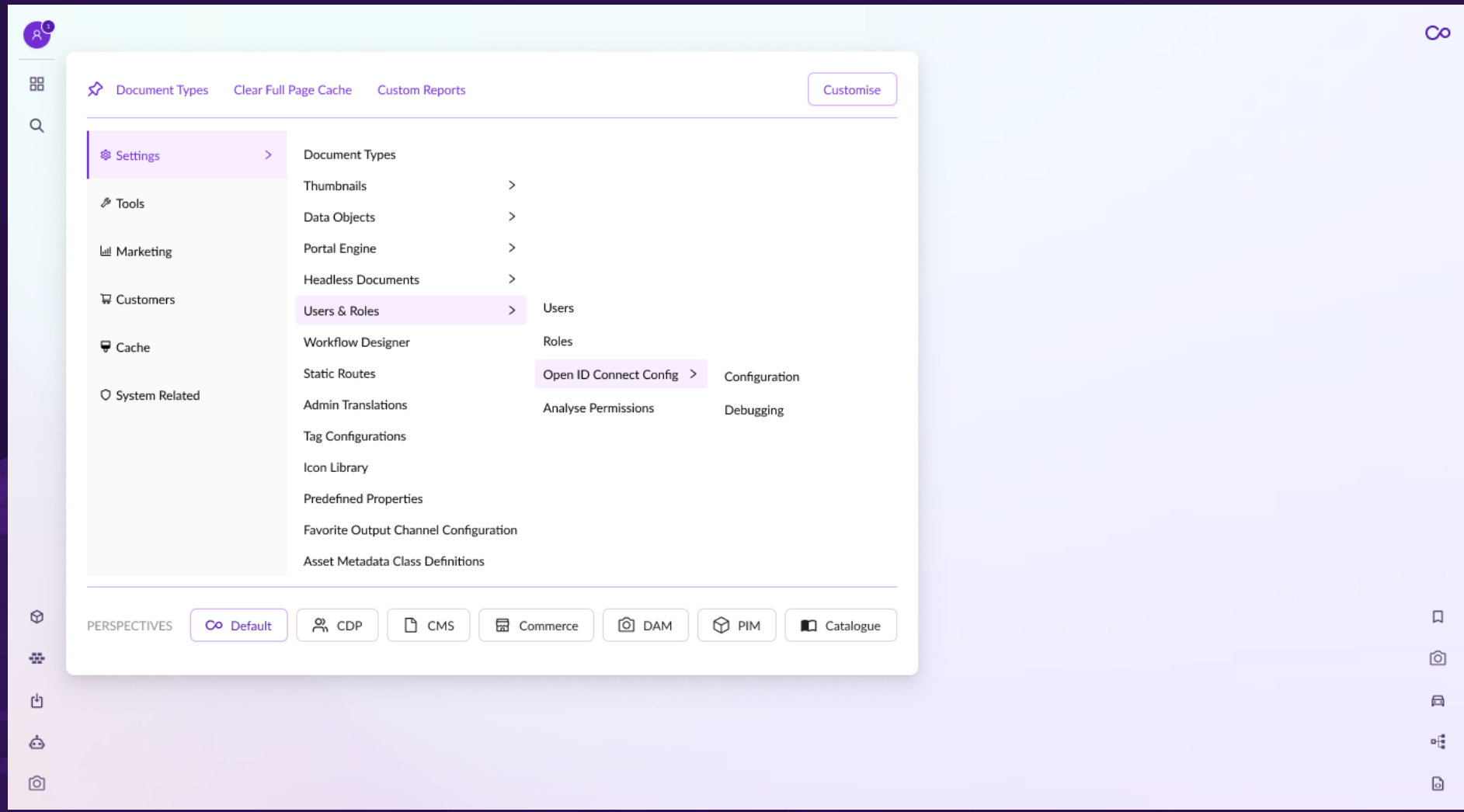
SCALABILITY



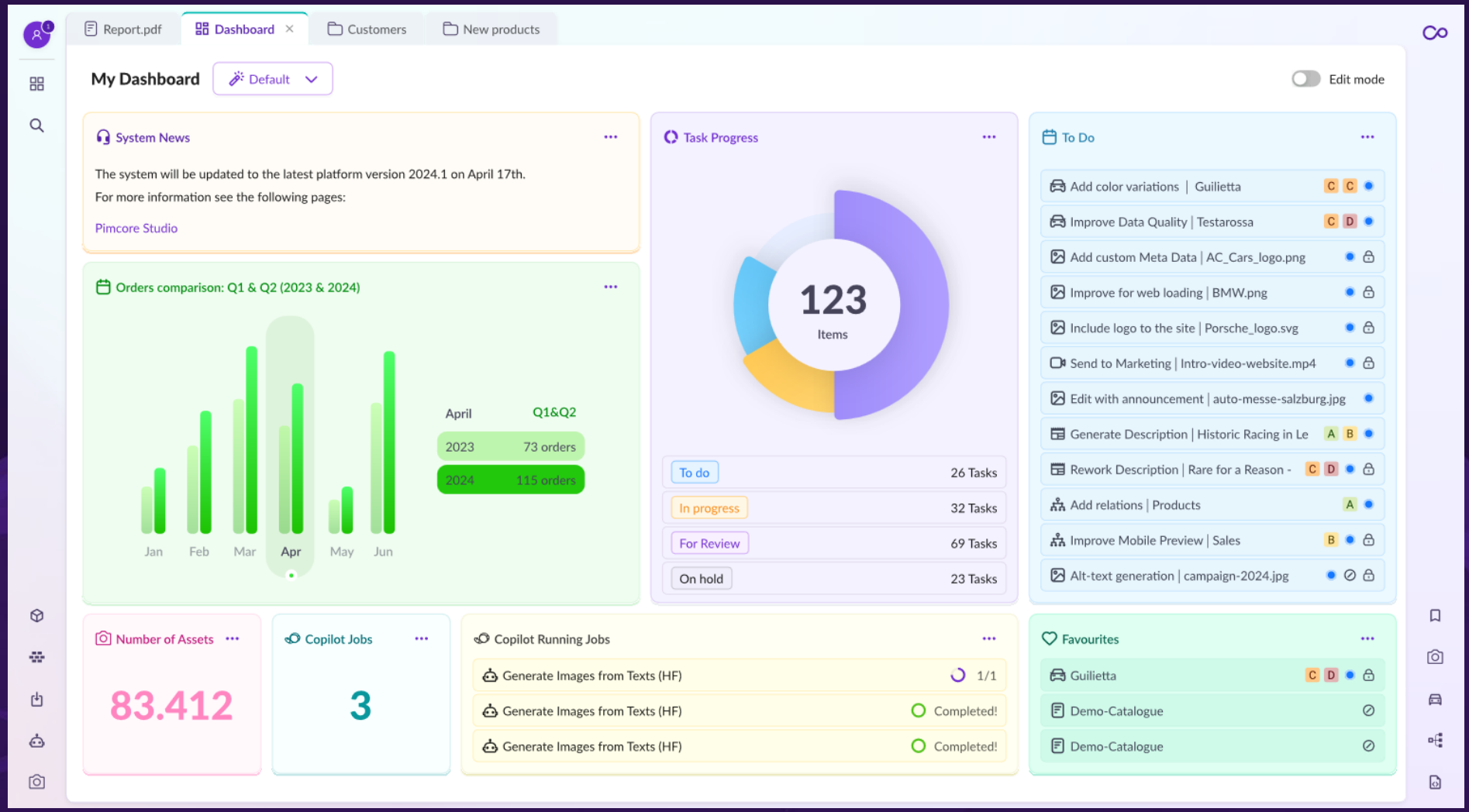
Studio Focus: Modern look & feel



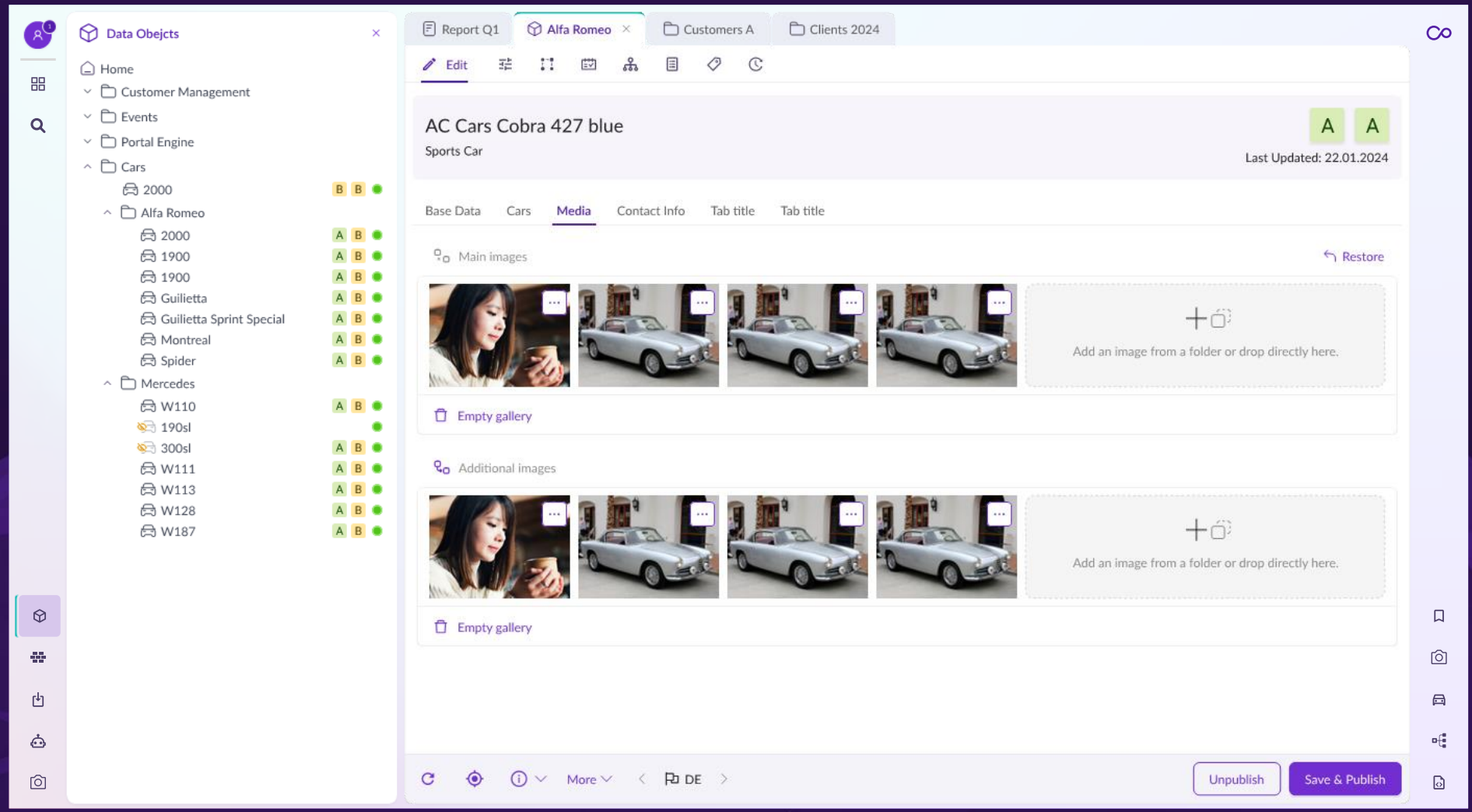
Studio Focus: Modern look & feel



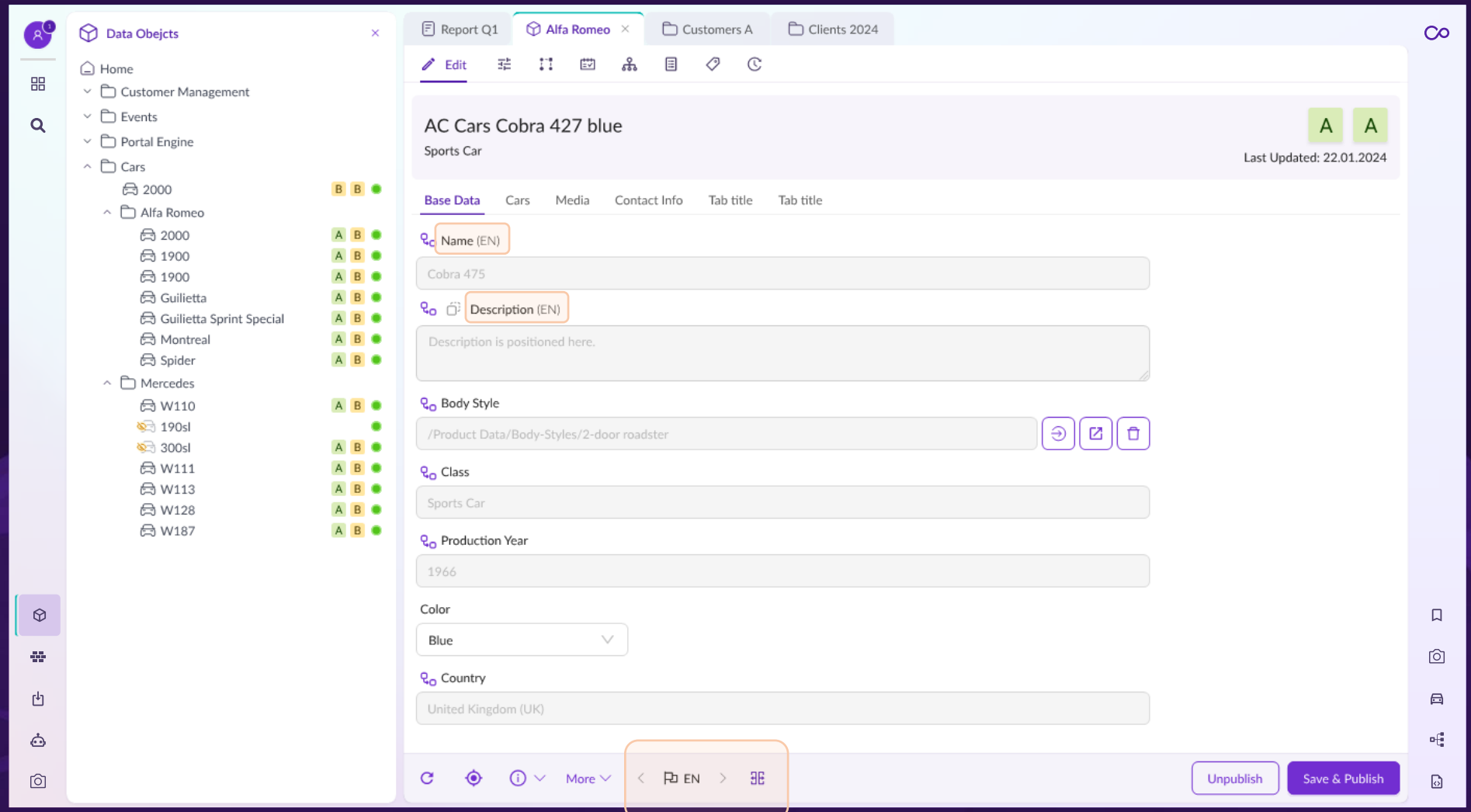
Studio Focus: Modern look & feel



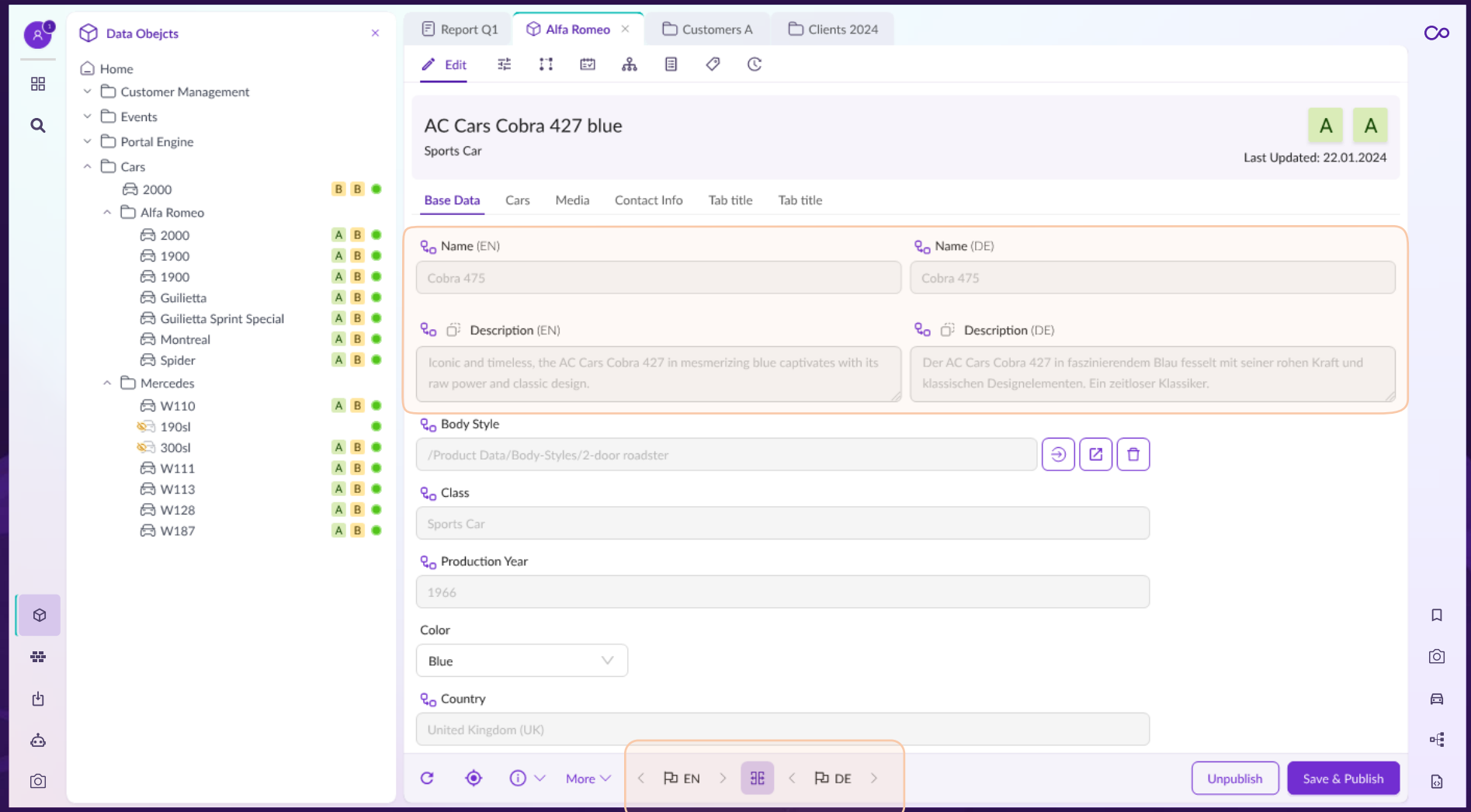
Studio Focus: Clean & Simple Design



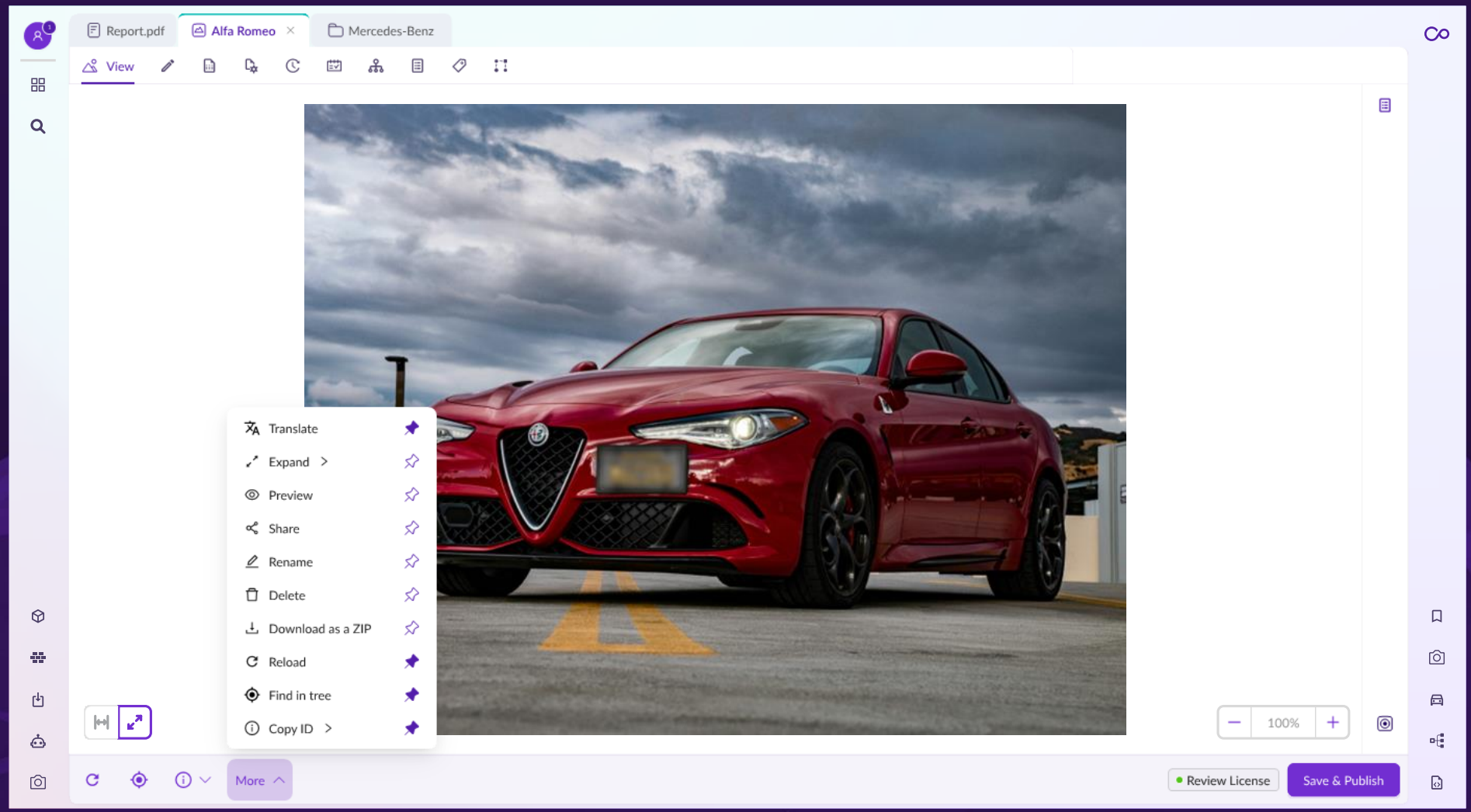
Studio Focus: Clean & Simple Design



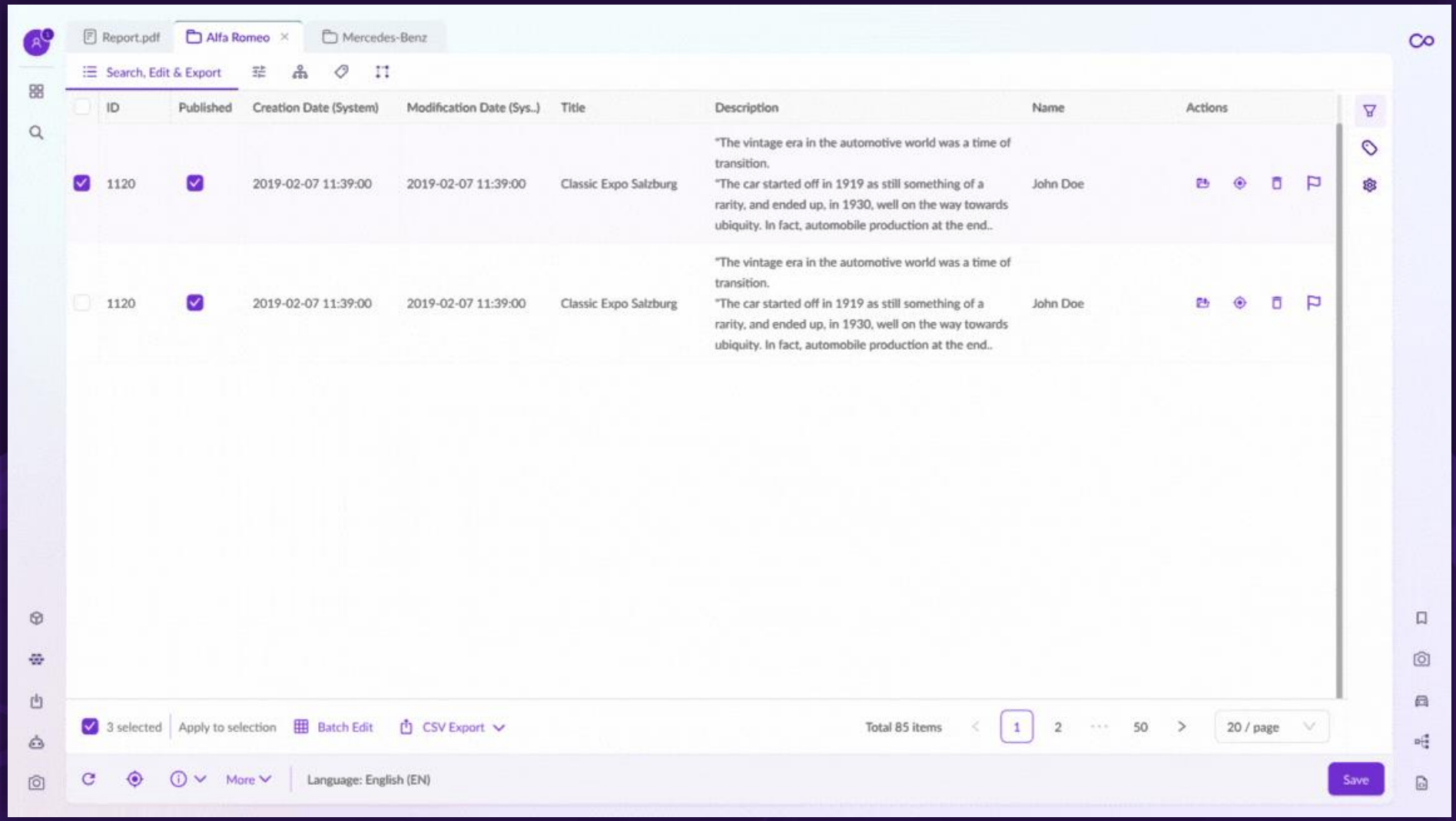
Studio Focus: Clean & Simple Design



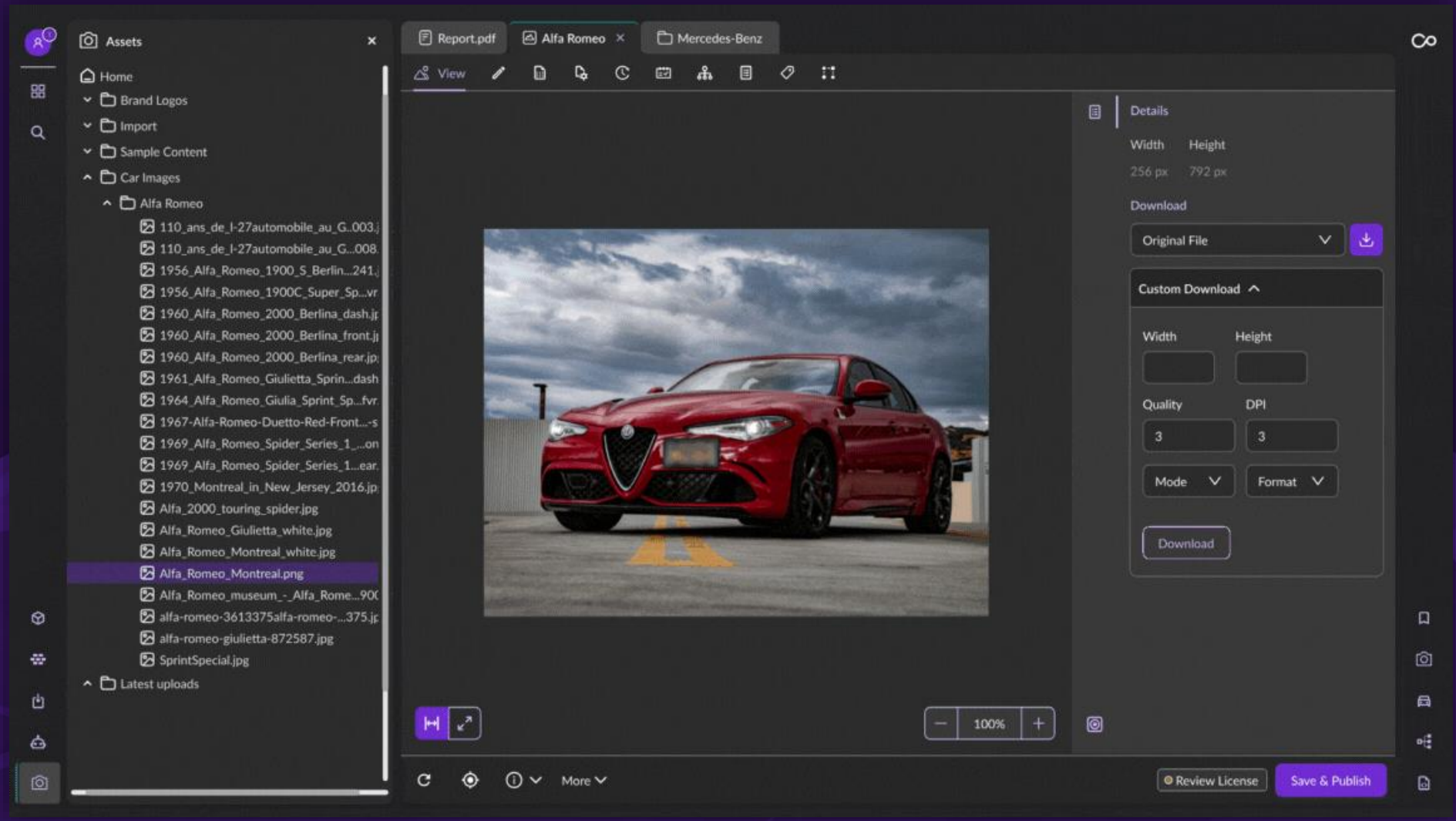
Studio Focus: Clean & Simple Design

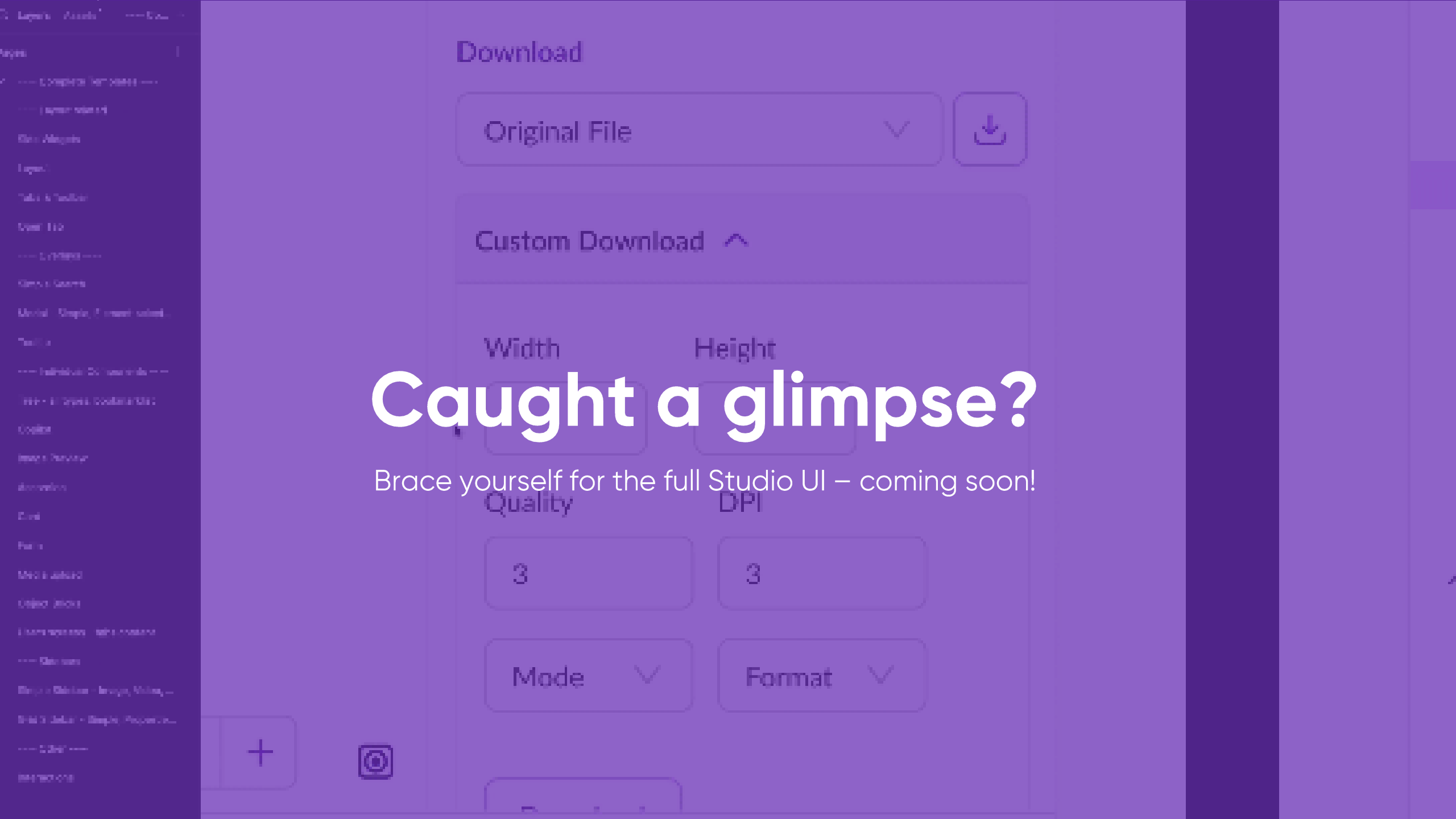


Studio Focus: Consistency in Concepts



Studio Focus: Theming



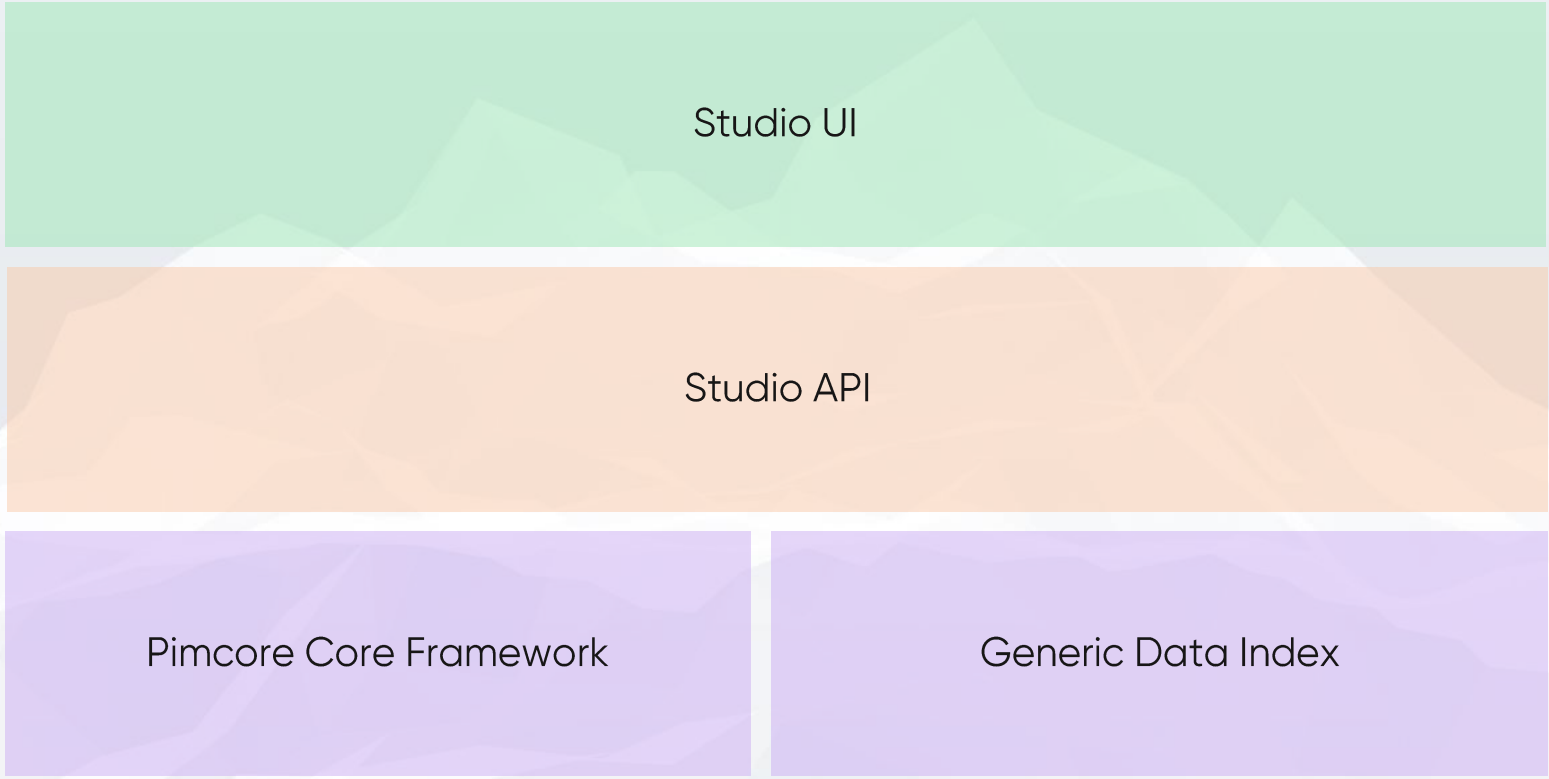


Caught a glimpse?

Brace yourself for the full Studio UI – coming soon!

Implementation

Components


Studio UI
Studio API
Pimcore Core Framework
Generic Data Index

Generic Data Index

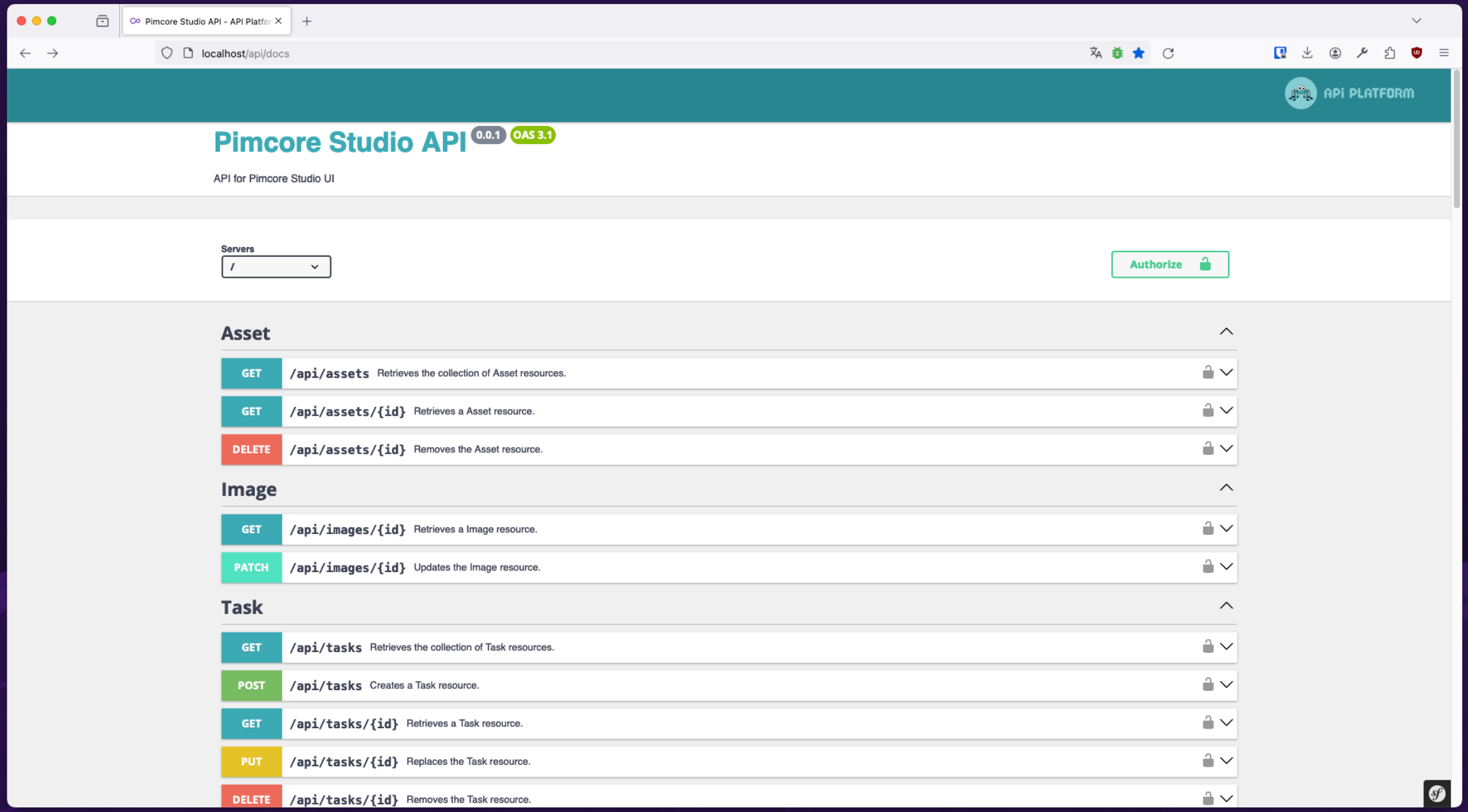
Generic Data Index

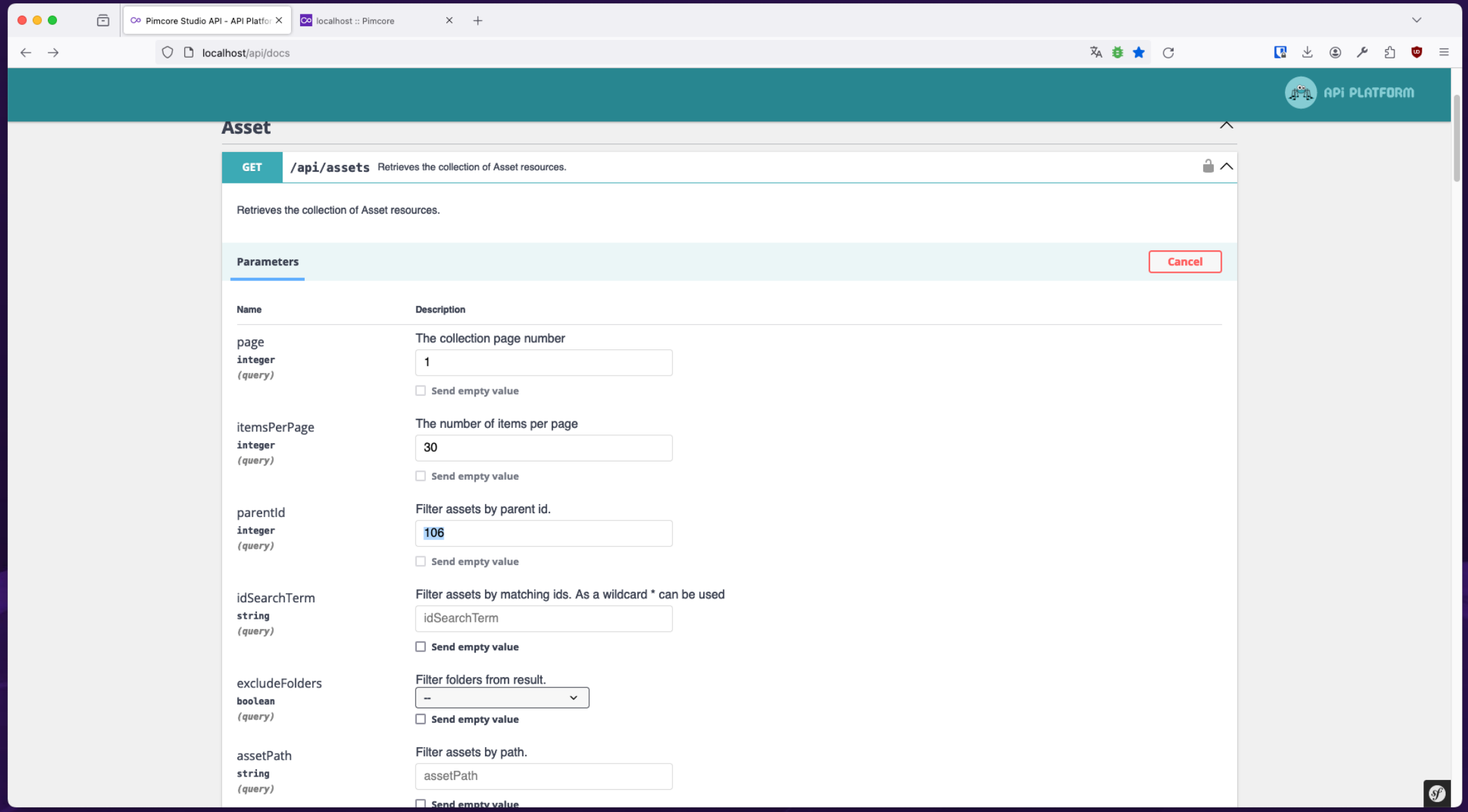
What	Why	How
One Generic Data Index	Improve Performance and Search Result Quality	Based on OpenSearch
Used by different Applications	Simplify Filtering for Data Precondition Services PQL	Search Services
Index Management works Out-Of-The-Box	No individual Indices	Configurable & Extendable

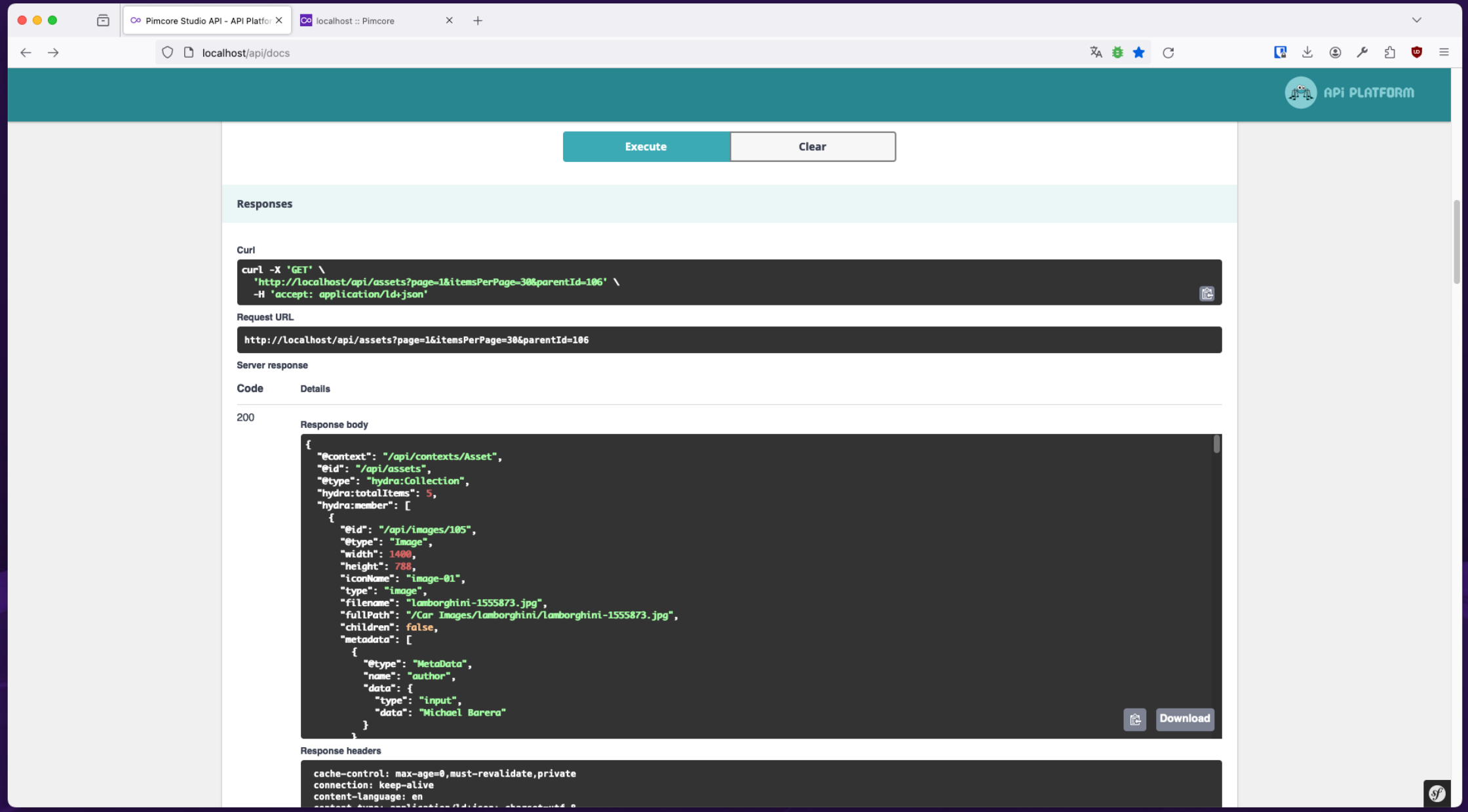
Studio API

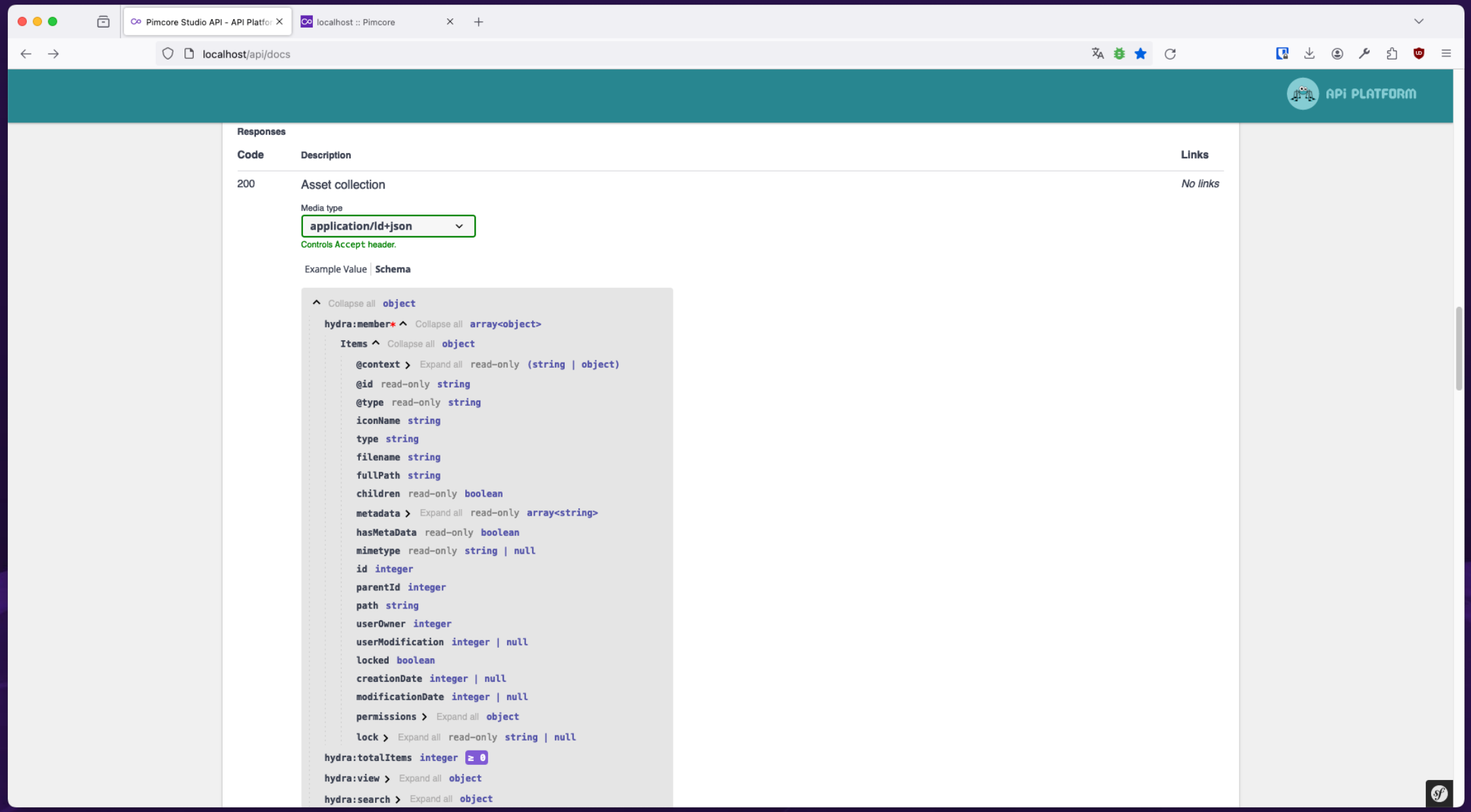
Studio API

What	Why	How
Primary focus on Studio UI	Previous API tightly coupled to extJS	API-Platform
Covers all Admin UI Tasks	Standards like JSON-LD, Hydra and OpenAPI	REST where feasible
Third party extensions	Generate client code automatically	Filters, pagination, partial updates
Not for headless frontend applications (Datahub)		


Pimcore Studio API - API Platfor
localhost/api/docs
API PLATFORM
Pimcore Studio API 0.0.1 OAS 3.1
API for Pimcore Studio UI
Servers
/
Authorize
Asset
GET /api/assets Retrieves the collection of Asset resources.
GET /api/assets/{id} Retrieves a Asset resource.
DELETE /api/assets/{id} Removes the Asset resource.
Image
GET /api/images/{id} Retrieves a Image resource.
PATCH /api/images/{id} Updates the Image resource.
Task
GET /api/tasks Retrieves the collection of Task resources.
POST /api/tasks Creates a Task resource.
GET /api/tasks/{id} Retrieves a Task resource.
PUT /api/tasks/{id} Replaces the Task resource.
DELETE /api/tasks/{id} Removes the Task resource.


Pimcore Studio API - API Platfor
localhost :: Pimcore
localhost/api/docs
API PLATFORM
Asset
GET
/api/assets
Retrieves the collection of Asset resources.
Retrieves the collection of Asset resources.
Parameters
Cancel
Name
Description
page
integer
(query)
The collection page number
1
Send empty value
itemsPerPage
integer
(query)
The number of items per page
30
Send empty value
parentId
integer
(query)
Filter assets by parent id.
106
Send empty value
idSearchTerm
string
(query)
Filter assets by matching ids. As a wildcard * can be used
idSearchTerm
Send empty value
excludeFolders
boolean
(query)
Filter folders from result.
--
Send empty value
assetPath
string
(query)
Filter assets by path.
assetPath
Send empty value


Pimcore Studio API - API Platfor
localhost :: Pimcore
localhost/api/docs
API PLATFORM
Execute
Clear
Responses
Curl
curl -X 'GET' \
'http://localhost/api/assets?page=1&itemsPerPage=30&parentId=106' \
-H 'accept: application/ld+json'
Request URL
http://localhost/api/assets?page=1&itemsPerPage=30&parentId=106
Server response
Code
Details
200
Response body
{
"@context": "/api/contexts/Asset",
"@id": "/api/assets",
"@type": "hydra:Collection",
"hydra:totalItems": 5,
"hydra:member": [
{
"@id": "/api/images/105",
"@type": "Image",
"width": 1400,
"height": 788,
"iconName": "image-01",
"type": "image",
"filename": "lamborghini-1555873.jpg",
"fullPath": "/Car Images/lamborghini/lamborghini-1555873.jpg",
"children": false,
"metadata": [
{
"@type": "MetaData",
"name": "author",
"data": {
"type": "input",
"data": "Michael Barera"
}
Download
Response headers
cache-control: max-age=0,must-revalidate,private
connection: keep-alive
content-language: en

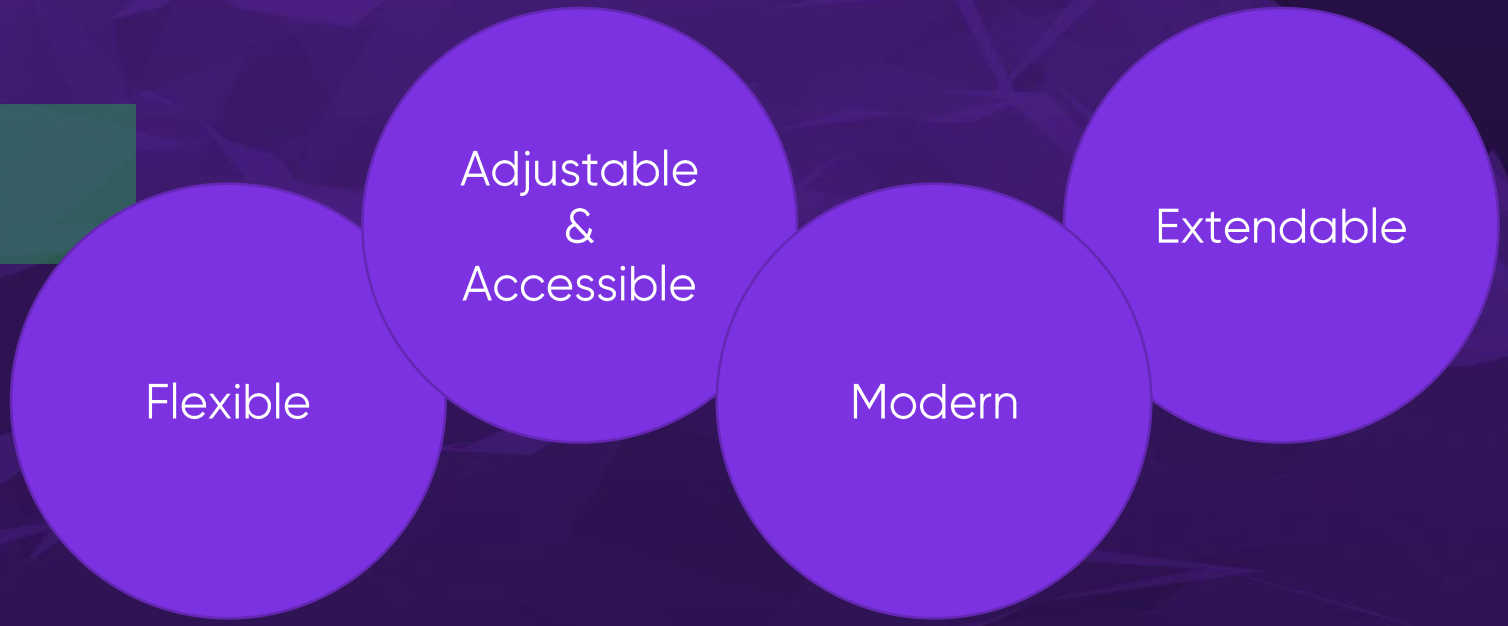

Pimcore Studio API - API Platfor
localhost :: Pimcore
localhost/api/docs
API PLATFORM
Responses
Code
Description
Links
200
Asset collection
No links
Media type
application/ld+json
Controls Accept header.
Example Value
Schema
Collapse all object
hydra:member* Collapse all array<object>
Items Collapse all object
@context Expand all read-only (string | object)
@id read-only string
@type read-only string
iconName string
type string
filename string
fullPath string
children read-only boolean
metadata Expand all read-only array<string>
hasMetaData read-only boolean
mimetype read-only string | null
id integer
parentId integer
path string
userOwner integer
userModification integer | null
locked boolean
creationDate integer | null
modificationDate integer | null
permissions Expand all object
lock Expand all read-only string | null
hydra:totalItems integer ≥ 0
hydra:view Expand all object
hydra:search Expand all object

Studio UI

Studio UI - What

New Admin Backend UI
for Pimcore

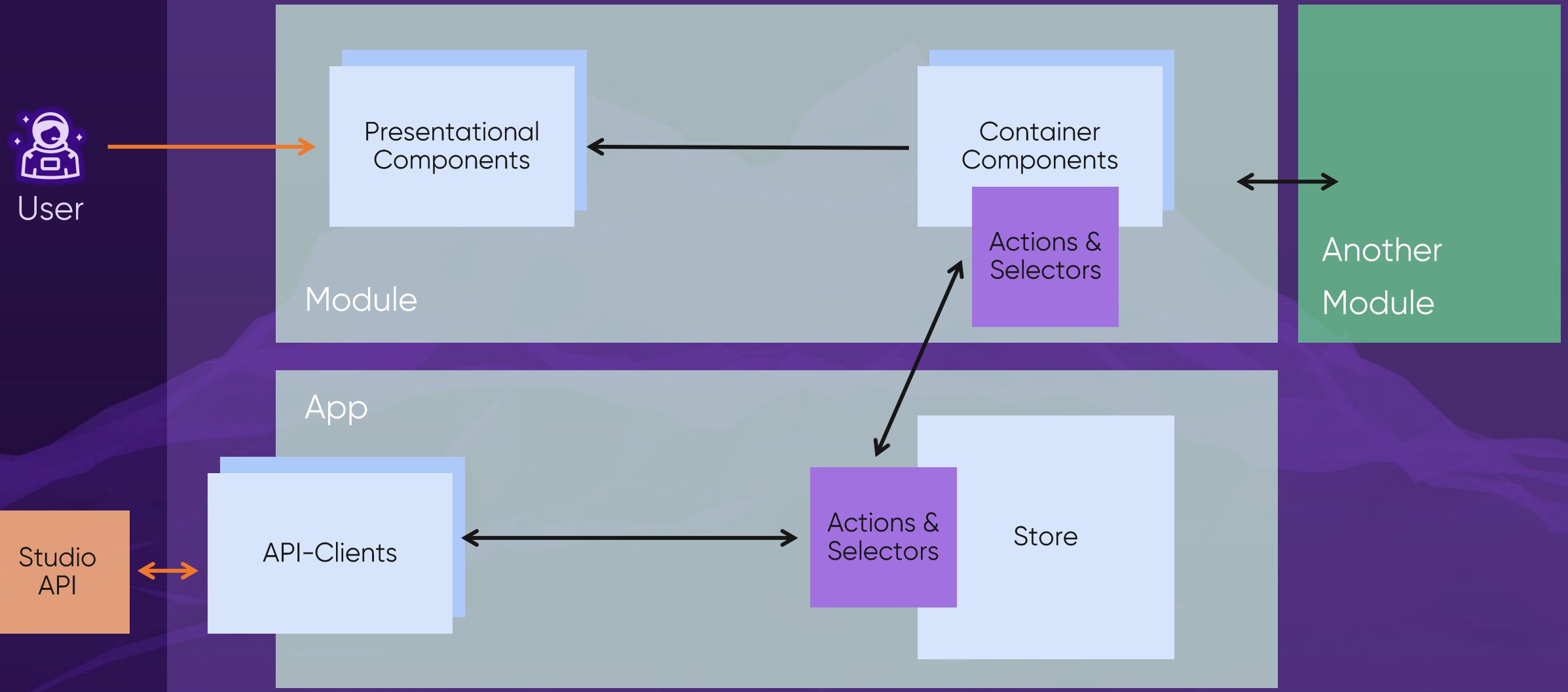
F.A.M.E



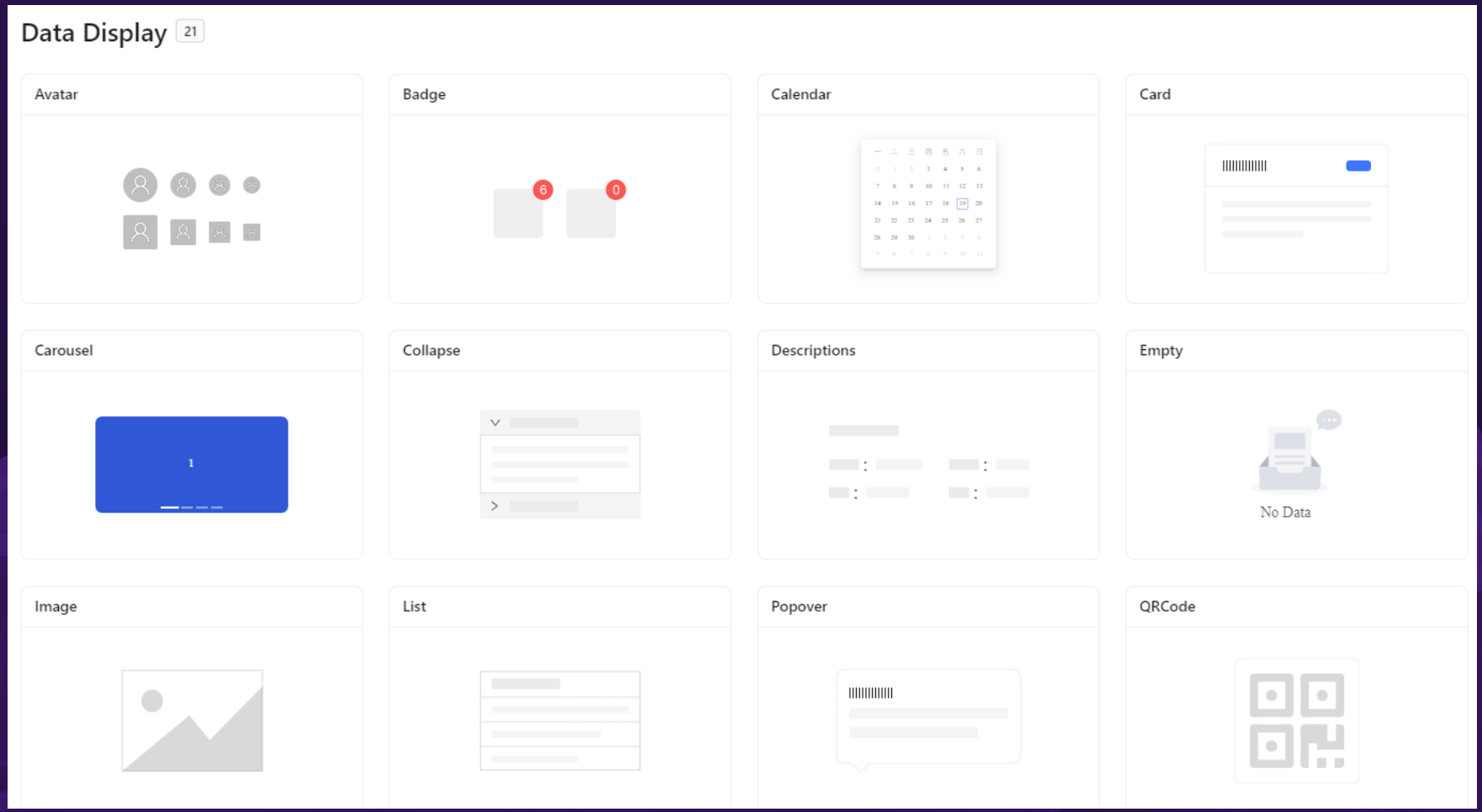
Studio UI - How

Technolgy stack	Other Tools
React Redux Ant Design	TypeScript Webpack / Symfony Encore Storybook ESLint Jest

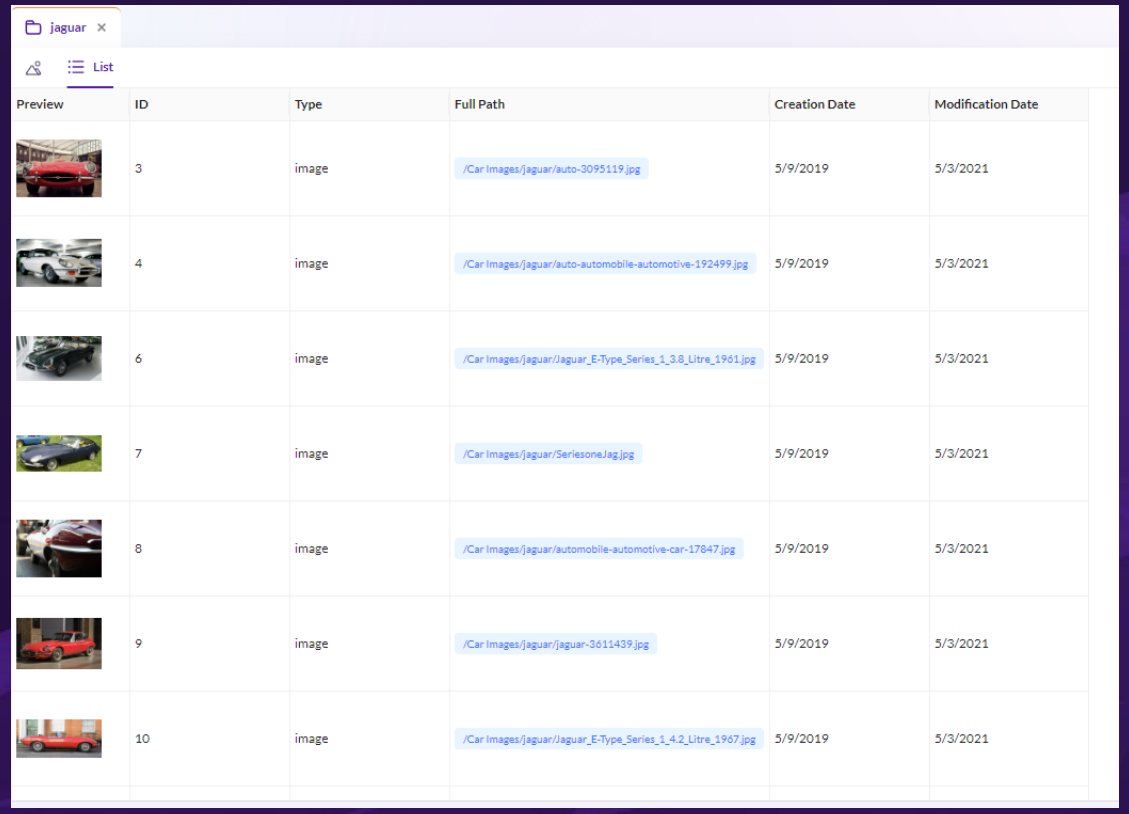
Studio UI - Architecture



Studio UI – Ant Design Components

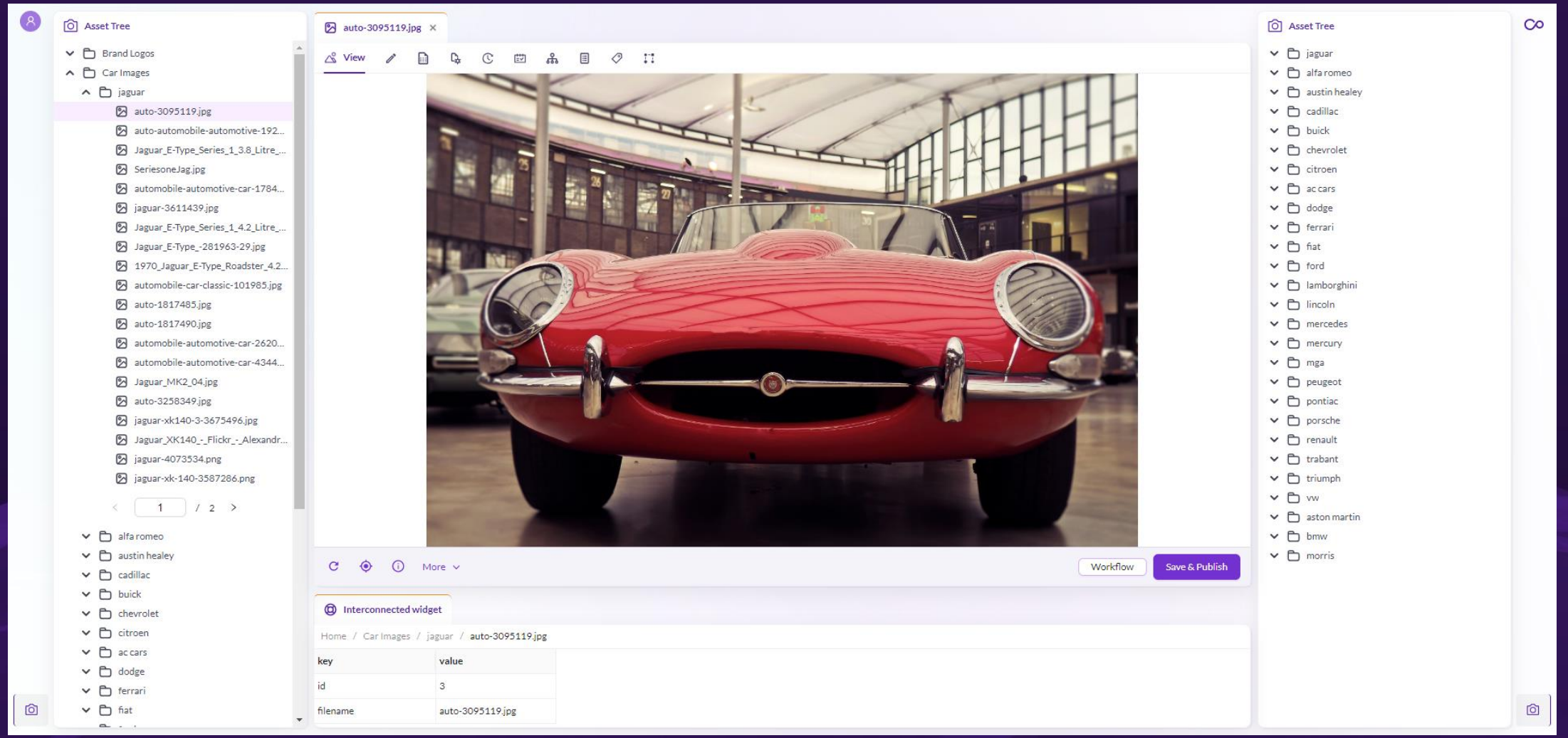


Studio UI – Grid



- Build on top of TanStack Table
- Powered by Ant Design styles

Studio UI – Widget Manager



Live Demo

Studio UI – How to extend and customize

What

Low-level apis

Reusable Components

Custom hooks

Theming

Documentation

Studio UI – Theming

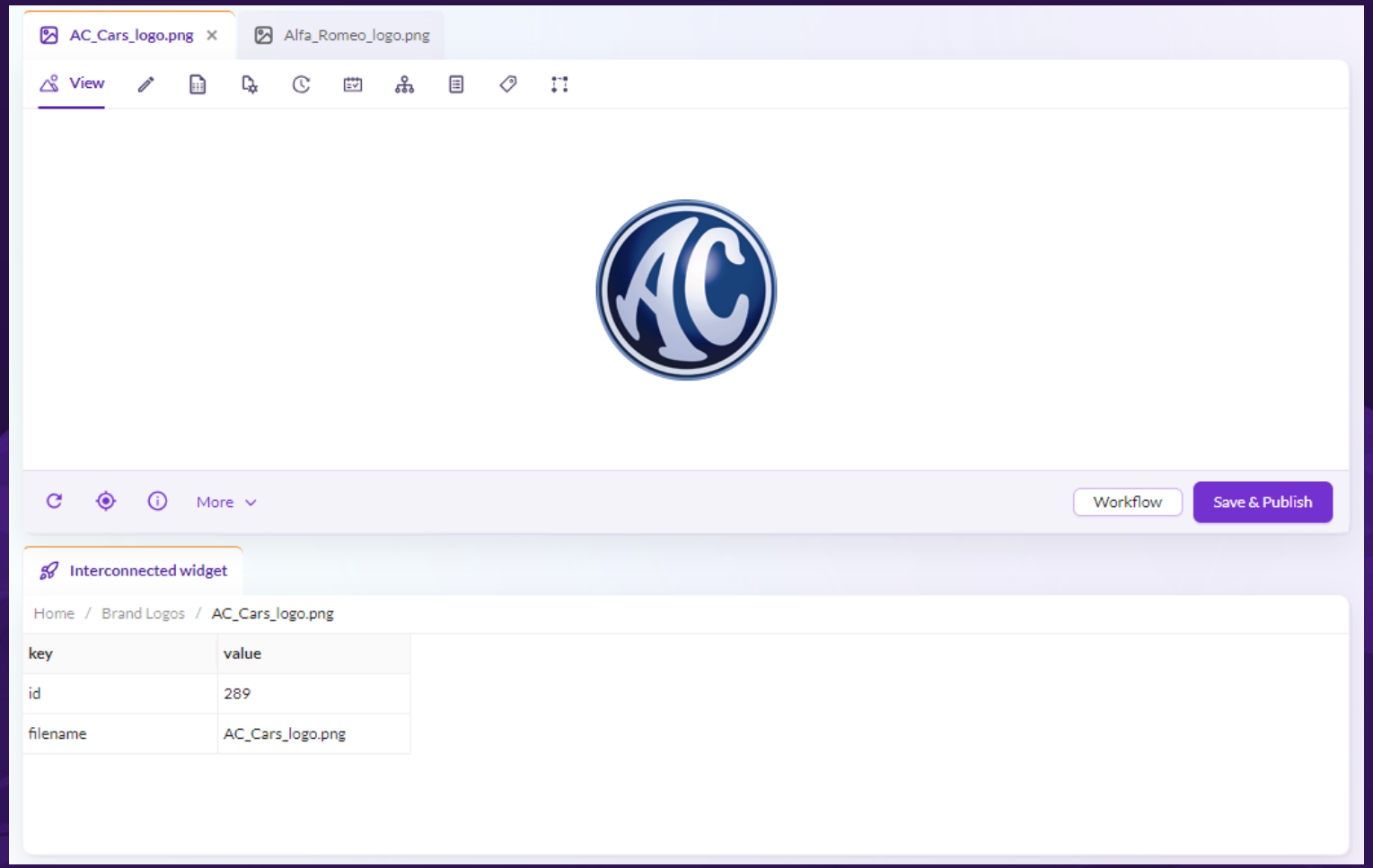
What

Ant Design powered design tokens

```tsx
<AntdThemeProvider
  theme={{
    token: {
      colorTextTreeElement: '#404655',
    }
  }}
>
  {children}
</AntdThemeProvider>
```

```tsx
export const useStyles = createStyles(({ token, css }) => {
  return {
    tree: css`
      padding: ${token.paddingXXS}px 0 ${token.paddingXS}px 0;
      max-width: 100%;
      color: ${token.colorTextTreeElement}
    `
  }
})
```

Studio UI – Example



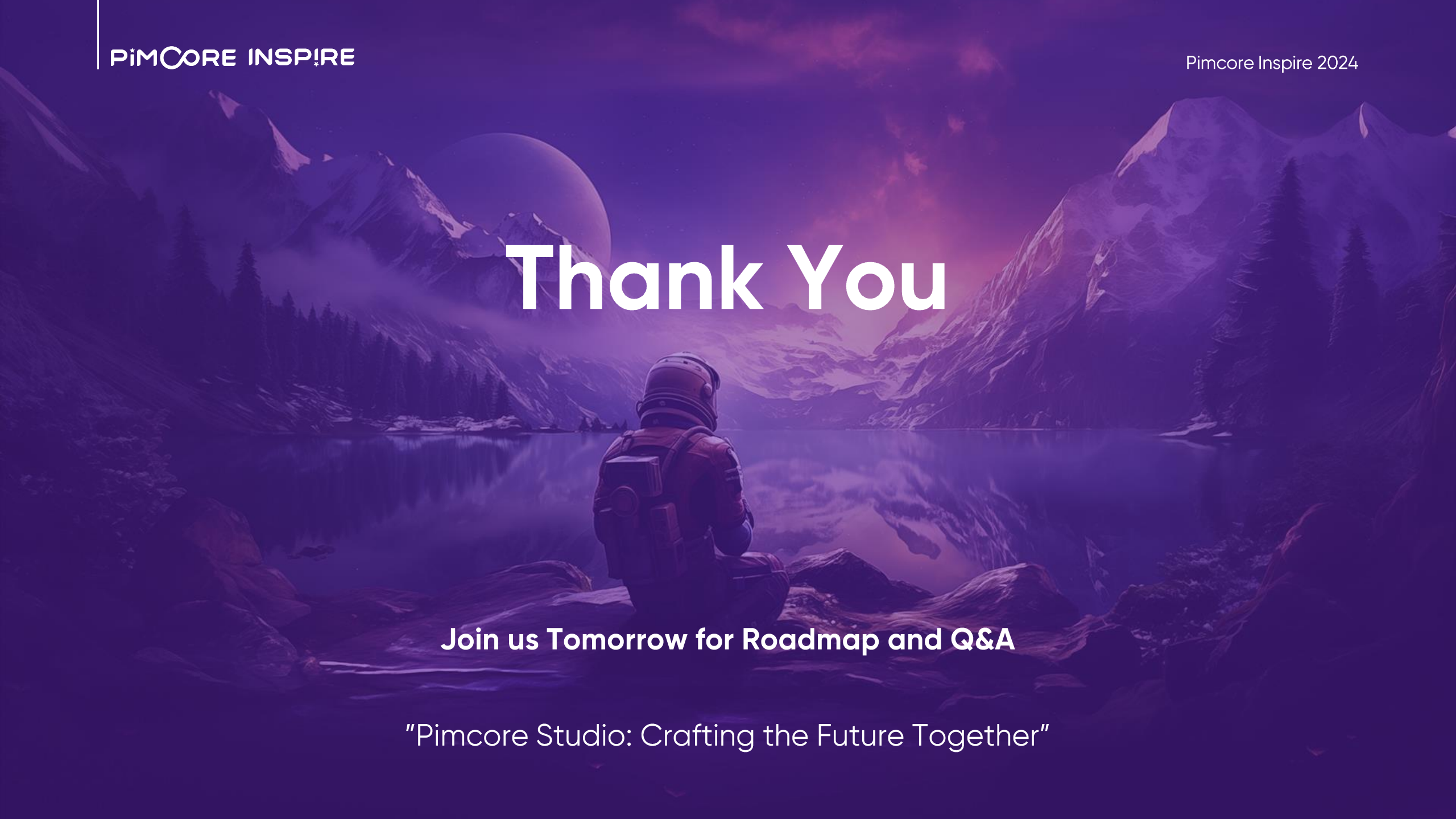
Studio UI – Example

```
registerWidget({
  name: 'example',
  component: Example
})
```

```
export const Example = (): React.JSX.Element => {
  const { context } = useGlobalAssetContext()

  if (context === undefined) {
    return <Result title="No context" />
  }

  return (
    <AssetContainer id={context.config.id} />
  )
}

```

Studio UI – Example

```tsx
export const AssetContainer = ({ id }: AssetContainerProps): React.JSX.Element => {
  const { isLoading, isError, data } = useApiAssetsIdGetQuery({ id: id.toString() })

  if (isError) {
    return <div>Error</div>
  }

  if (isLoading || data === undefined) {
    return <div>Loading...</div>
  }

  const assetData = [
    { key: 'id', value: data.id },
    { key: 'filename', value: data.filename }
  ]

  const columnHelper = createColumnHelper()

  const columns = [
    columnHelper.accessor('key', {}),
    columnHelper.accessor('value', {})
  ]

  return (
    <Grid data={assetData} columns={columns} />
  )
}
```



Thank You

Join us Tomorrow for Roadmap and Q&A

"Pimcore Studio: Crafting the Future Together"