

Addon

 Language: [Deutsch](#) · [English](#)

Addon

Extend the web client with addons – layout components, dashboard, background functions and the addon API.

KONZEPT

DEVELOPER

UPDATED: JUL 2026

APPLIES TO NUCLOS 4.2026.X

On this page

- [Advantages](#)
- [Use cases](#)
- [Accessing the context](#)
- [Related pages](#)

With **addons**, the web client can be extended with (project-specific) functions that should not become part of the Nuclos core.

Advantages

- leaner web client, better performance;
- easily removable – no legacy code;
- clearly defined interfaces foster modularity;
- customer-specific extensions and use of non-open-source libraries possible.

Use cases

- **layout addon**: new GUI components in the layout of a detail form;
- addons in the dashboard or the result list;
- (hidden) background functions reacting to user interactions (similar to Groovy rules for calculated attributes in the Java client).

Accessing the context

Via a defined API, addons access data and functions of the web client, e.g. the layout context:

Fehler beim Rendern des Makros 'code': Ungültiger Wert für den Parameter 'com.atlassian.confluence.ext.code.render.InvalidValueException'

```
@Inject(LAYOUT_CONTEXT) private layoutContext: LayoutContext;  
// ...  
this.layoutContext.getAddonProperty('someProperty');
```



Development

The development environment for addons is described in [JS web client development environment](#) (package @nuclos/nuclos-addon-api).

Related pages

JS web client development environment

Environment.

[Open](#)

