Jira key:	Reporter:	Story points:
RAIN-290	Radu Cosnita	13

Summary:

HIP registry meta model

Description

As PSS / Third party developer I want to know what is the bundle meta model so that I can register RAIN components easily.

Acceptance

- A document describing the bundle meta model will be presented
- CloudIA / RAIN will agree on the bundle meta model.
- Examples of how register / unregister requests will look like.

Jira key:	Reporter:	Story points:
RAIN-202	Radu Cosnita	21

Summary:

RAIN OAUTH2 client research

Description

As PSS / Third party developer I want RAIN security to be integrated with OAUTH2 so that authentication / authorization is done transparently for me. It means that RAIN must have an OAUTH2 client that can be used to fulfill OAUTH2 1&1 flows. Here you need to collaborate with CloudIA team to understand all the OAUTH2 flows that are / will be implemented. Check https://github.com/ciaranj/node-oauth to see an example of node oauth consumer project.

Acceptance

- A feature proposal for OAUTH2 RAIN client will be presented.
- The login flow with OAUTH2 enabled must be presented (doc).
- The data store flow with OAUTH2 enabled must be presented (doc).

Jira key:	Reporter:	Story points:
RAIN-117	Radu Cosnita	13

Summary:

RAIN client side text translations.

Description

As PSS / Third party developer I want to access the localized text resources from client side controller so that I can do client side text translation.

Acceptance

- The code must be commited on rainjs.
- A sample application must be commited on rainjs so that client side text translation is proved.
- The login form from previous sprints must be translated to en_US and de_DE (use google translate). The login form will be shown in both locales.
- Unit tests must be written for success / failure scenarios.
- Documentation must be written for PSS / Third party developers to prove how client side text translation works.
- The sample application must be presented from dev environment.

Jira key:	Reporter:	Story points:
RAIN-110	Radu Cosnita	8

Summary:

RAIN localization service

Description

As RAIN I will provide a service to return modules text resources in json format so that PSS / Third party developers can localize client side texts. The service will retrieve the po catalog content for the specified language and will give message ids and message text under a json representation. If the locale specified is not found the default platform locale will be retrieved but a warning message must be logged. See also [RAIN-100]

Acceptance

- An integration test must be presented.
- Unit tests for success / fallback / failure scenarios.
- Documentation must be written for PSS / Third party developers so that they know how to obtain texts for client side localization.
- The integration test must be presented for dev environment.

Jira key:	Reporter:	Story points:
RAIN-10	Radu Cosnita	13

Summary:

RAIN datagrid component.

Description

As a Developer / Third party developer I want to use a filterable and sortable table so that I present large set of data in a friendly manner. This must be a reusable table component with Webdesk look & feel (as UCD is not yet available). The table must provide filtering, sorting and searching features. In addition no performance penalties are accepted (the table must react fast for large data sets > 300 records). We must not write the component from scratch. We will choose an existing and tested component. [RAIN-105]

More information can be found on:
http://styleguide.schlund.de/konzeption/guidelines/wtg1/designpattern_workplace/interaction_tables.htm

You can find the attached information within the issue attachments.

Acceptance

- A sample application must be created to show how the component can be used. The headers must be localized in en_US and de_DE format. Example must emphasize support for UTF-8 special characters.
- Documentation must be written for PSS / Third party developers (how to / examples).
- The sample application must be presented from dev machine.