Nova Web Application – Inner component communication

# The reason.

We need to find a way how to communicate between different component inside Nova application and how we can track/notify about changes in different models (application parts)

Some examples:

* when user try to modify any part of artifact - try to lock artifact/change status in the header when locked/undo all changes, refresh artifact if client version is behind server version/undo all changes, refresh artifact, change status to locked by different user and make read-only when artifact already locked by different user on the server;
* when user add and post comment in Discussion panel we want comment indicator appear in the Project explorer tree node for corresponding artifact;
* when user add/remove attachment in Utility panel we want '\*' (change) indicator appear in the artifact header and node text in the Explorer become blue...

# ReactiveX

Reactive Extensions (Rx) is a library for composing asynchronous and event-based programs using observable sequences and LINQ-style query operators.

See details: [http://reactivex.io/](http://reactivex.io/rxjs/)

# Key features

[Observable](https://github.com/Reactive-Extensions/RxJS/blob/master/doc/api/core/observable.md) - a data source that can be observed, meaning that it can send data to anyone who is interested, it can be subscribe to, where the value coming down is either what is provided by the source, or transformed by something.

[Observer](https://github.com/Reactive-Extensions/RxJS/blob/master/doc/api/core/operators/subscribe.md) - an object that acts as a handle for the subscription, i.e. to register an observer in the end to the values coming out of a stream.

[Subject](https://github.com/Reactive-Extensions/RxJS/blob/master/doc/api/subjects/subject.md) - a proxy that acts both as an observer and as an Observable. Because it is an observer, it can subscribe to one or more Observables, and because it is an Observable, it can pass through the items it observes by reemitting them, and it can also emit new items.

# Installation

See instruction in README.md file in a root of NovaWeb application

# Sample of usage

Current implementation (June 2016) of Project Explorer component provides access to three subjects: ProjectCollection, CurrentProject, CurrentArtifact.

… in Project-Manager.ts

this.projectCollection = new Rx.BehaviorSubject<Models.IProject []>([]);

this.currentProject = new Rx.BehaviorSubject<Models.IProject>(null);

this.currentArtifact = new Rx.BehaviorSubject<Models.IArtifact>(null);

To publish values to the sequence call <subject>.onNext(T)

this.projectCollection.onNext(this.projectCollection.getValue());

…

this.currentArtifact.onNext(artifact);

…

this.currentProject.onNext(project);

When subject is created we can subscribe to these subject to listen the value to be publish

… in Project-Explorer.ts

//subscribe for current artifact change (need to distinct artifact)

this.projectManager.projectCollection.subscribeOnNext(this.onLoadProject, this),

this.projectManager.currentArtifact.distinctUntilChanged().subscribeOnNext(this.onSelectArtifact, this),

Read more….

* [RxJS Design Guidelines](http://xgrommx.github.io/rx-book/content/guidelines/)
* [The introduction to Reactive Programming](https://gist.github.com/staltz/868e7e9bc2a7b8c1f754)
* <http://reactivex.io/rxjs/>
* <https://github.com/Reactive-Extensions/RxJS>
* <https://github.com/Reactive-Extensions/RxJS-DOM>