An Infusion Pattern Language

Parts to be Replaced

Document expected grade options in code

How It Works

Grade options are specified with null values (or in the case of invokers, fluid.notImplemented may be used), optionally with a comment describing the role of the option, or who is expected to provide the value. These grade options document expectations of code that will work with components of the grade.

Example: kettle.middleware

The kettle.middleware grade documents the middleware invoker interface, a single invoker called "handle", using fluid.notImplemented:

```
// The base middleware grade defines a function accepting the request object and returning a promise
fluid.defaults("kettle.middleware", {
    gradeNames: ["fluid.component"],
    invokers: {
        handle: {
            funcName: "fluid.notImplemented"
        }
    }
});
```

https://github.com/fluid-project/kettle/blob/master/lib/KettleMiddleware.js

Marker Grade

Use grade names to annotate components

How It Works

A marker grade is a grade, typically with no implementation, that is used to annotate a component as having a particular role or position in the component tree. The marker grade may then be used, for example, to locate the annotated component in IoC expressions.

Example: gpii.nexus.componentRootHolder

The Nexus Co-Occurrence Engine uses a marker grade gpii.nexus.componentRootHolder to indicate the parent of the Nexus "component root":

```
fluid.defaults("gpii.nexus.componentRootHolder", {
    gradeNames: ["fluid.component"]
});
```

This marker grade is used within the Co-Occurrence Engine distributeOptions rule:

```
distributeOptions: [
    {
        target: "{gpii.nexus.componentRootHolder gpii.nexus.componentRoot fluid.component}.options.listeners",
        record: {
            "onCreate.fireCoOccurrenceEngineComponentCreated":
                "{gpii.nexus.coOccurrenceEngine}.events.onComponentCreated",
            "afterDestroy.fireCoOccurrenceEngineComponentDestroyed":
                "{gpii.nexus.coOccurrenceEngine}.events.onComponentDestroyed"
        },
        namespace: "coOccurrenceEngine"
    }
]
```

In this case, the marker grade provides flexibility in the component structure that the Co-Occurrence Engine is participating in. The gpii.nexus.componentRootHolder need only be findable by the Infusion IoC system. For example, the component root holder could be a sibling, a parent, or a grandparent of the Co-Occurrence Engine.

https://github.com/simonbates/co-occurrence-engine/blob/master/src/CoOccurrenceEngine.js

Example: kettle.middlewareHolder

Kettle uses a marker grade kettle.middlewareHolder to indicate containers of middleware

https://github.com/fluid-project/kettle/blob/master/lib/KettleMiddleware.js

Distribute Options

When to Use It

The component options being distributed are specified at a distance from the component(s) that will be affected. This quality may make it hard to see the source of all of the configuration options that a component will receive.

Examples

- Kettle configurations of the GPII use distributeOptions to inject configuration and sub-components into parts of the system
    - https://github.com/GPII/universal/blob/master/gpii/configs/gpii.config.cloudBased.flowManager.production.json
- The Nexus Co-Occurrence Engine uses distributeOptions to inject listeners into components that have not been created yet
    - https://github.com/simonbates/co-occurrence-engine/blob/master/src/CoOccurrenceEngine.js

Pipeline

Structure tasks in a pipeline using fluid.promise.fireTransformEvent

How It Works

A pipeline organizes an activity into multiple ordered stages, with the output of one stage being fed as the input to the next stage.

The fluid.promise.fireTransformEvent function may be used to build pipelines in Infusion.

When to Use It

See also: fluid.promise.sequence

Wrap Third-Party APIs for Use in Infusion

Examples

- gpii-pouchdb: https://github.com/GPII/gpii-pouchdb/blob/master/src/js/pouchdb-common.js