adaptTo()

APACHE SLING & FRIENDS TECH MEETUP

BERLIN, 22-24 SEPTEMBER 2014

Apache Sling Generic Validation Framework

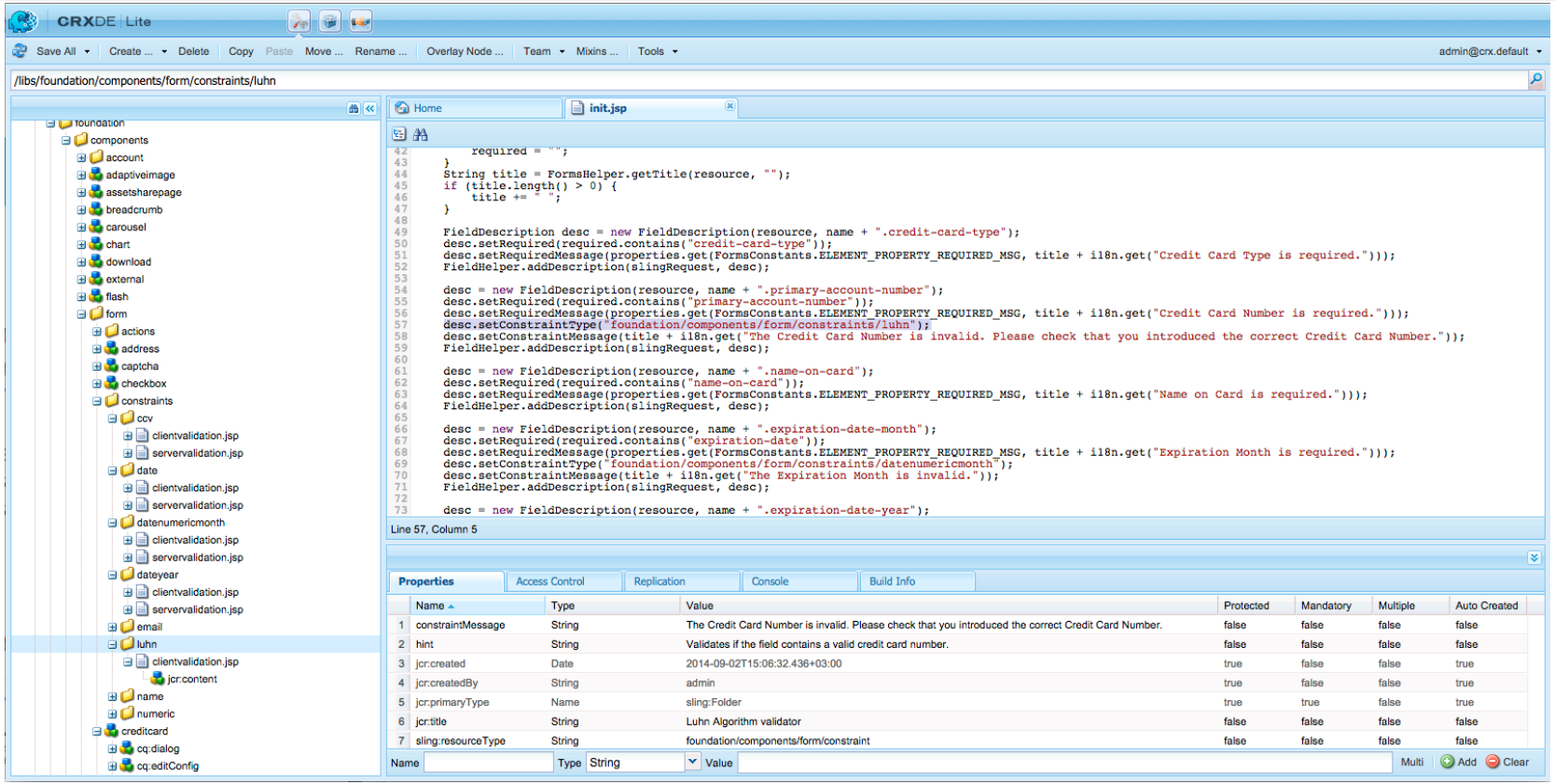
Radu Cotescu
ASF committer, Sling contributor
Computer Scientist @ Adobe Systems
@raducotescu - radu@apache.org

Adobe

Apache Sling Generic Validation Framework

The most common web application security weakness is the failure to properly validate input from the client or environment. [1]

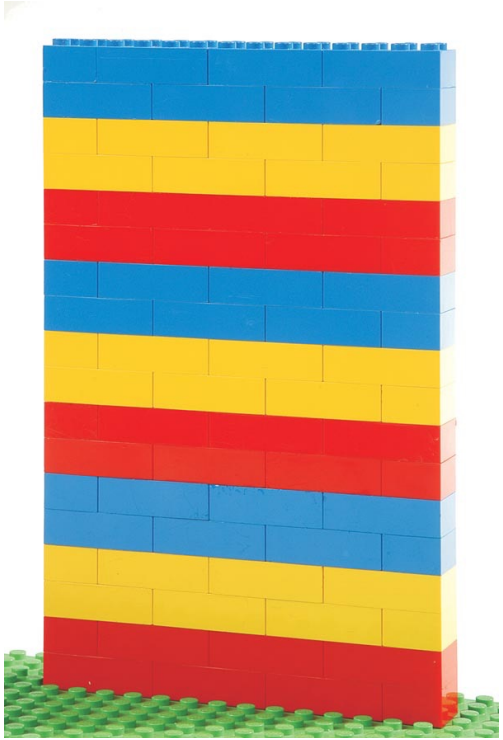
A Bit of History



A Bit of History

The Building Blocks

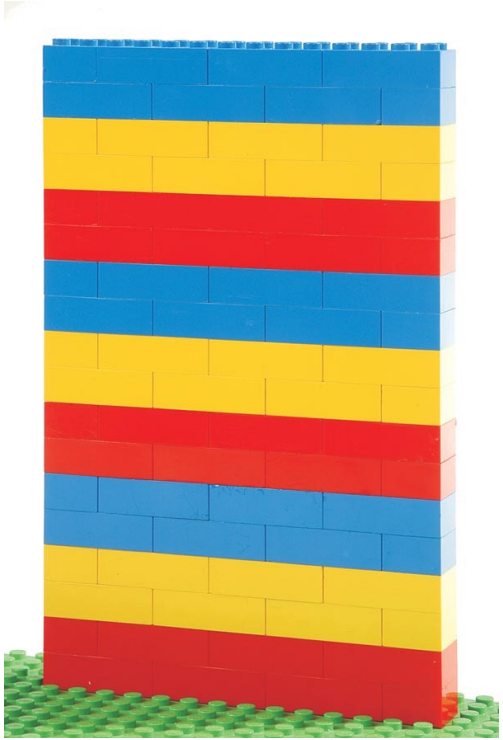
The Building Blocks



`ValidationService`

- main entry point into the Validation API
- responsible for retrieving a `ValidationModel` and for performing the validation operation

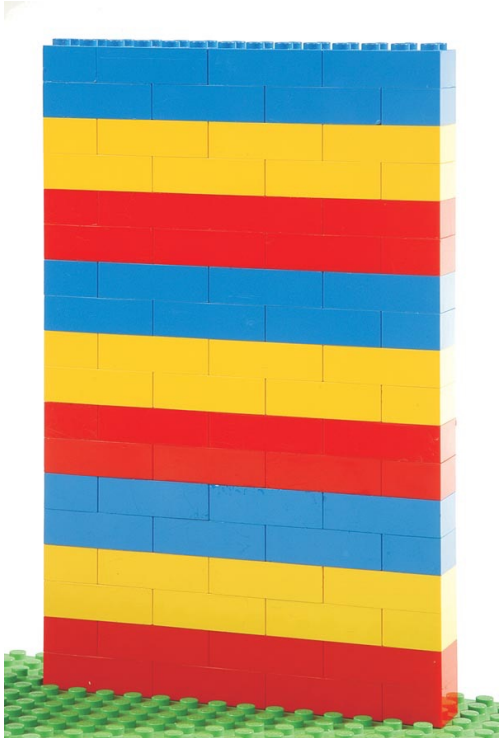
The Building Blocks



`ValidationModel`

- descriptive structure for the validated object

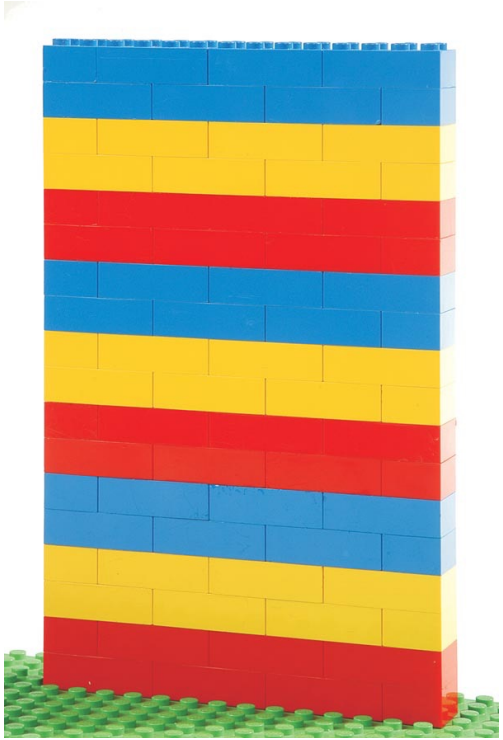
The Building Blocks



`ResourceProperty`

- describes one of the validated object's properties
- it has a `Type` and optionally a `Validator`

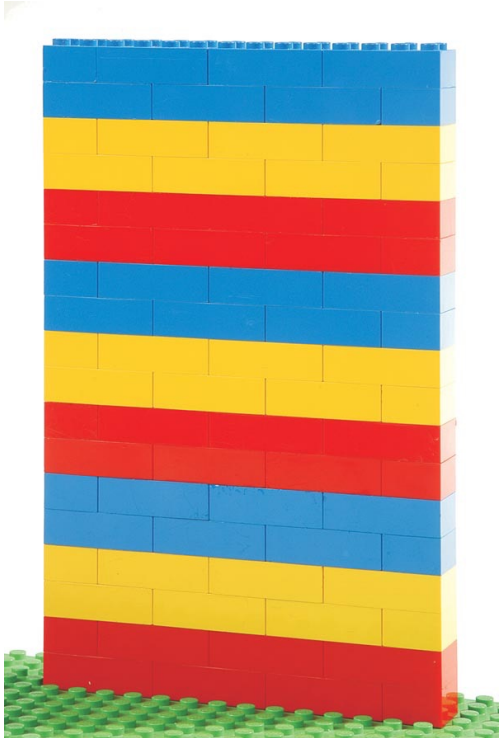
The Building Blocks



`Validator`

- validates a single piece of information
- can receive arguments

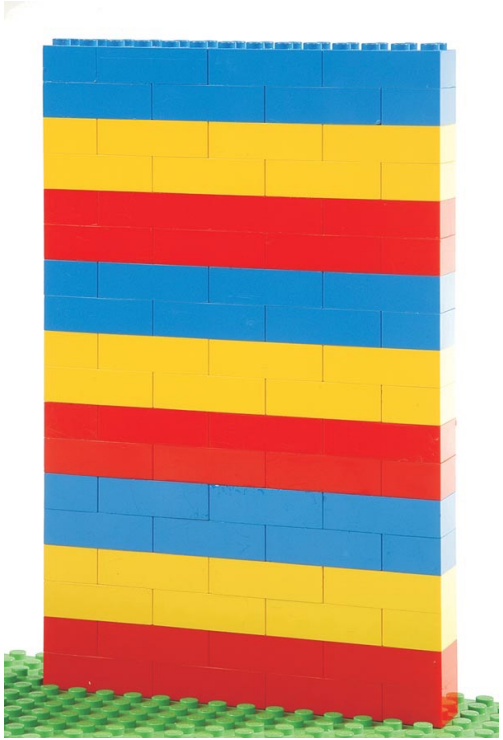
The Building Blocks



`ChildResource`

- defines validation rules for resource trees
- it's comprised of one or more `ResourceProperty` objects

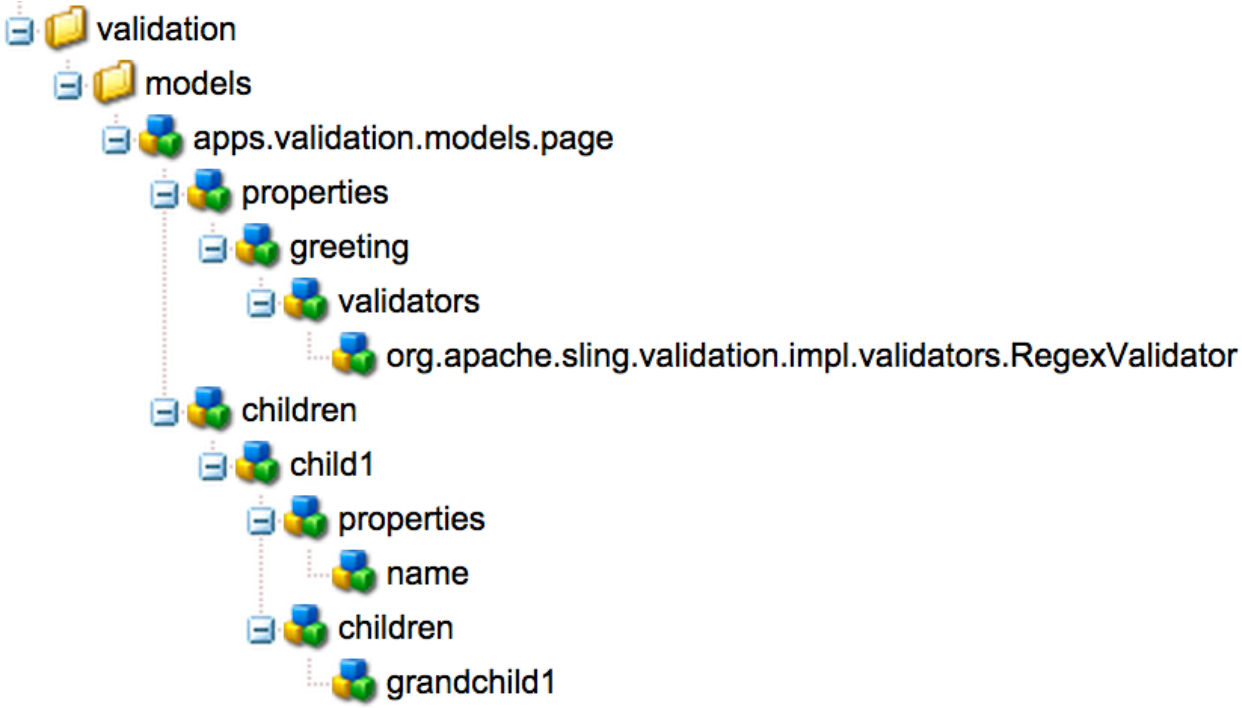
The Building Blocks



`ValidationResult`

- holds the validation result - boolean
- it can contain validation error messages

Expressing a `ValidationModel` as content



Expressing a `ValidationModel` as content

```
apps.validation.model.page
--applicablePaths=[‘/content/p/1’, ‘/content/p/2’]
--sling:resourceType=sling/validation/model
--validatedResourceType=/apps/p/c/page

greeting
--propertyType=string

org.apache.sling.validation.impl.validators.RegexValidator
--validatorArguments=[‘regex=^HelloWorld$’]
```

codeExamples();

The ValidationService

```
// resource validation
ValidationModel model = validationService.getValidationModel(resource);
if (model != null) {
  ValidationResult result = validationService.validate(resource, model);
}

// request validation
ValueMap map = request.adaptTo(ValueMap.class);
ValidationModel model =
validationService.getValidationModel(VALIDATED_RESOURCE_TYPE, APPLICABLE_PATH);
if (model != null) {
  ValidationResult result = validationService.validate(map, model);
}
```

Simple integration with Sling Models

```
@PostConstruct
protected void validateResource() {
    ValidationModel vm = validationService.getValidationModel(resource);
    if (vm != null) {
        ValidationResult vr = validationService.validate(resource, vm);
        if (!vr.isValid()) {
            // do your processing here
        }
    }
}
```

Features available today

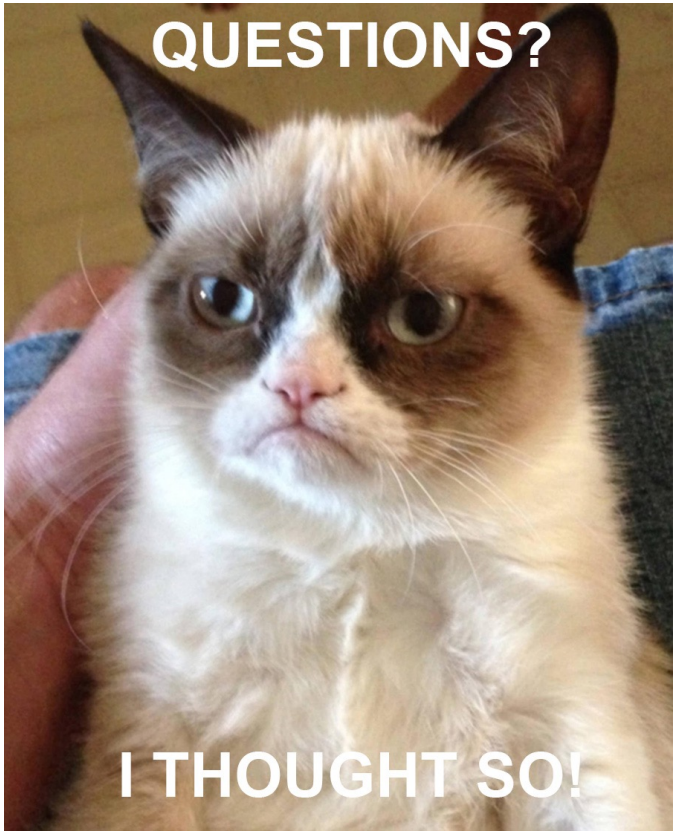
1. `ValidationModels` based on content structures.
2. The `ValidationModels` allow resource-tree validation but also request parameters validation.
3. Non-intrusive for existing Sling Models

Planned features

1. Provide JavaScript validators.
2. Translate `ValidationModel` content structures into JavaScript objects for client-side validation.
3. Define a Validation client library.

Demo

Apache Sling Generic Validation Framework



Thank you!

Links & Resources

https://github.com/raducotescu/org.apache.sling.validation
https://issues.apache.org/jira/browse/SLING-2803

[1] https://www.owasp.org/index.php/Data_Validation#Description

The demo artifacts can be found in the `examples` folder from the Git repository.