



adaptTo()

APACHE SLING & FRIENDS TECH MEETUP
2 - 4 SEPTEMBER 2019

Connect Apache Sling to Your Content

Karl Pauls & Carsten Ziegeler

intro()



- Computer Scientist @ Adobe
- Member of the Apache Software Foundation
- PMC Member of Apache Felix and Sling (VP Apache Felix)
- Co-Author OSGi in Action



- Principal Scientist @ Adobe
- Member of the Apache Software Foundation
- PMC Member of Apache Felix and Sling
- OSGi Expert Groups and Board member

main()

Apache Sling's Resource Abstraction

- A resource is a piece of content
 - Usually mapped to a JCR node
 - File system, bundle resource, database..
- Attributes of resources
 - Path in the resource tree
 - Resource type
 - Metadata, e.g. last modification date
 - ValueMap

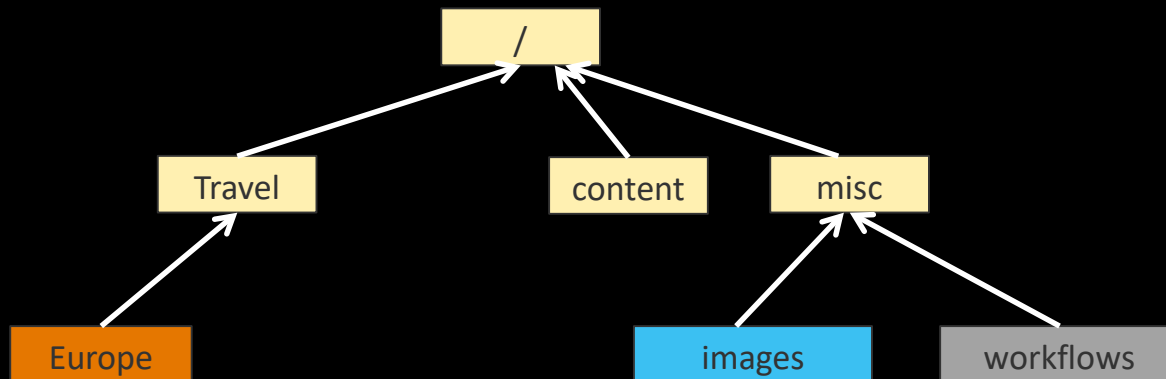
Resource Resolver

- Tasks:
 - Finding resources
 - Getting resources
 - Querying resources
 - CRUD operations
- Not Thread Safe!
 - Includes all objects fetched via resolver

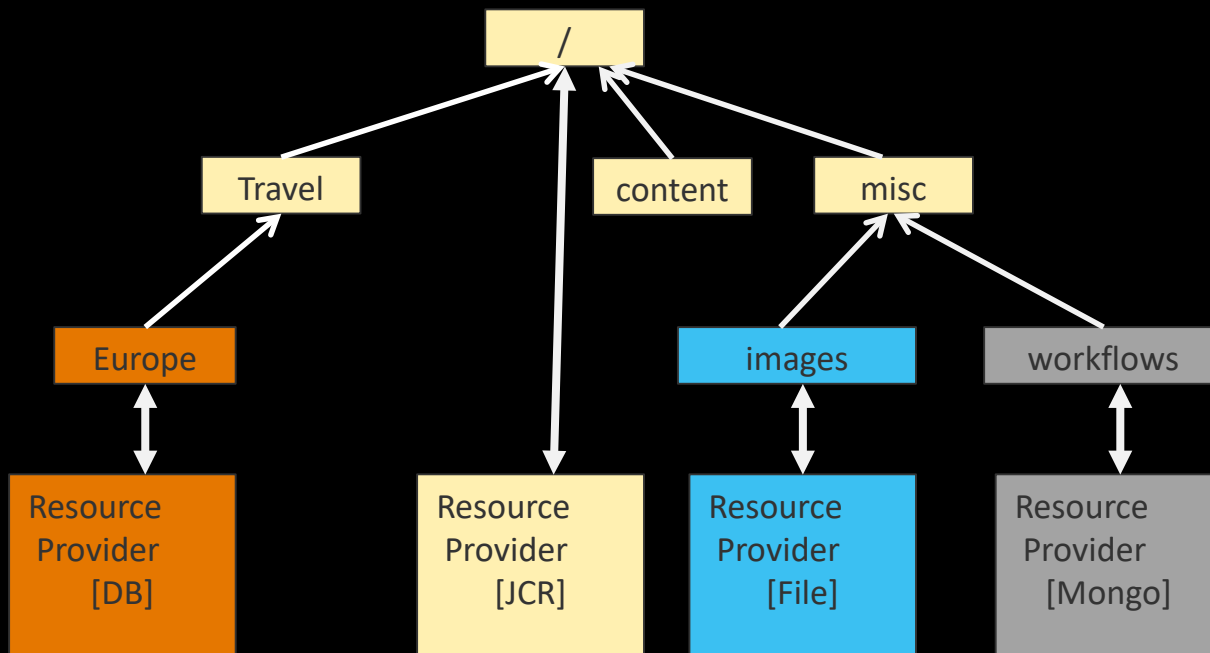
Where is Your Content?

- JCR
- Databases (SQL, NOSQL)
- Services (feeds...)
- Storage Providers

Resource Tree



Mounting Content in the Resource Tree



Implementing ResourceProvider

- Authentication (optional)
- Reading: two methods + Resource impl
- CRUD: seven simple methods
- Optional: query, observation

Code Example

```
@Component(  
    service = {ResourceProvider.class}  
)  
public class MyResourceProvider extends ResourceProvider {  
    @Override  
    public Resource getResource(ResolveContext resolveContext,  
        String path, ResourceContext resourceContext, Resource resource)  
    { ... }  
    @Override  
    public Iterator<Resource> listChildren(ResolveContext resolveContext,  
        Resource resource)  
    { ... }  
}
```

Demo

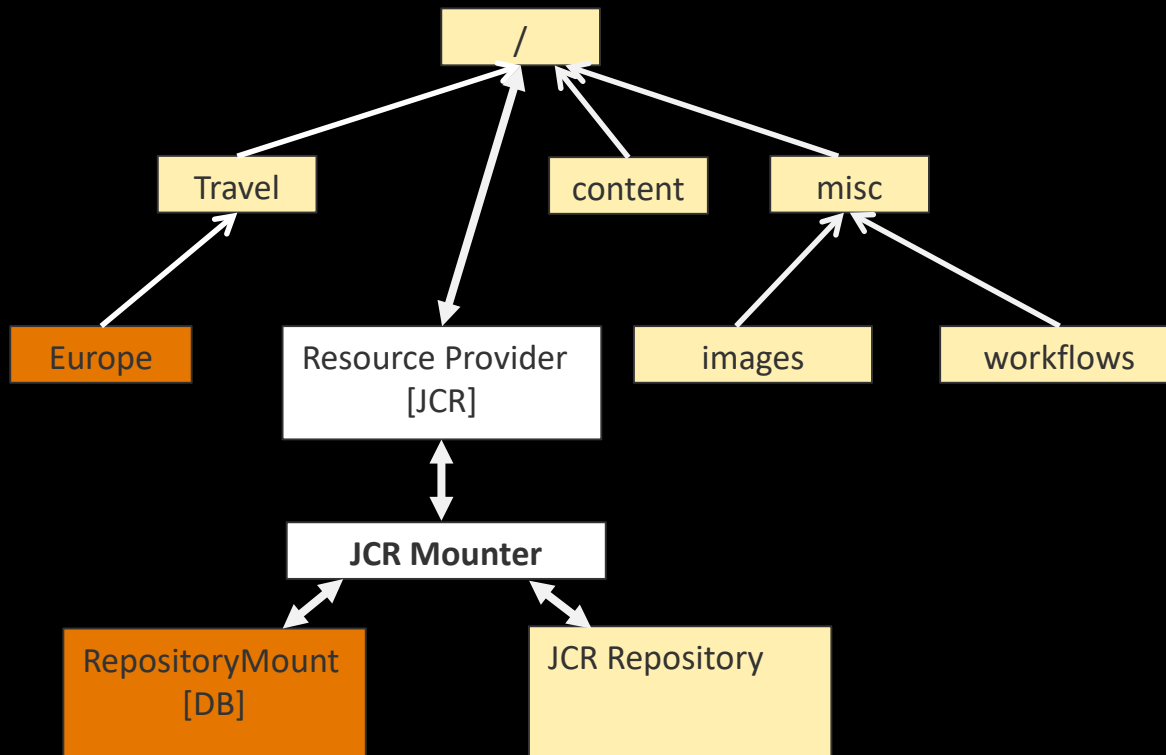
Connect Your Content using a ResourceProvider

- Nice, easy and straight forward
- Done!
- Done?

Potential Problems of using ResourceProvider

- Existing code might expect JCR Resource
- Existing code might only use JCR API
- Search often directly leveraging JCR search

Mounting Content using RepositoryMount



Implementing RepositoryMount

- Authentication (optional)
- Full JCR API
- Optional: query, observation

Code Example

```
public void doFilter(  
    ServletRequest pRequest,  
    ServletResponse pResponse,  
    FilterChain pChain) throws IOException, ServletException  
{  
    if (!(pRequest instanceof SlingHttpRequest)) {  
        throw new ServletException("Request is not a Sling HTTP request.");  
    }  
  
    Node node = ((SlingHttpRequest) request).getResource().adaptTo(Node.class);  
  
    ....  
}
```

Demo

Potential Problems of using RepositoryMount

- Complex and error prone
- Experimental
 - Still needs work to become more stable
 - Missing default implementations

- Verify your use case and code using your content!
- RepositoryMount or ResourceProvider
- Future Work:
 - Default implementation for remote resources

Q&A

Appendix

- Resource Provider

<https://sling.apache.org/documentation/the-sling-engine/resources.html#custom-resource-providers>

- RepositoryMount

<https://github.com/apache/sling-org-apache-sling-jcr-base/blob/master/README.md>

- Demo Code

<https://github.com/karlpauls/adapttorepositorymount>