

# Accessing Sling from a Fat Client

[phil.rice@softwareFm.org](mailto:phil.rice@softwareFm.org)



# Accessing Sling from a Fat Client

- Why Sling?
- Apache Http Client
- Authentication
- Flat hierarchy
- Data format
- Conclusion



## Why Jackrabbit?

- Structured and unstructured data
- Can handle large data volumes
- It's not a relational database
  
- It's an Apache project
- It's used by Adobe products



## Why Sling?

- Restful interface to Jackrabbit
- Nice mechanisms for injecting behaviour into server:
  - Aggregation of data on server
  - Logging
- It's an Apache project
- It's used by Adobe products



## Apache Http Client

- Very low level http client
- ThreadSafeClientConnManager
- See class ClientBuilder
- See class GetRequest



## Authentication

- Need some authentication to edit
- Basic authentication didn't work
- HttpClient allowed a RequestInterceptor
  - PreemptiveAuthInterceptor



## Flat Hierarchy

- I am storing information about:
  - Projects
  - Organisations
  - Jars
- Around  $10^5$  to  $10^7$  records
- Sling doesn't like  $10^7$  child nodes



## Flat Hierarchy

Naïve path

hostname/project/<projectName>

End up with  $10^7$  child nodes

Using a hash:

hostname/project/p37/<projectName>

Where 37 is the hash of <projectName>



## Data Format

- Using Json to “get” data from the server
  - `hostname/Project/p37/<projectName>.json`
- Domain model in Java is `Map<String, Object>`
- Changing the data using posts
- Type hints important
- Sometimes use the import operation



## Conclusions

- Sling:
  - Clean
  - Simple
  - Nice Edit/Compile/Debug
- Jackrabbit has many nice features but
  - The persistence layer doesn't scale with writes
  - The version API is hard to use

