

Services inside and out

Kyle Browning

What we are going to speak about?

- What is the services module and what does it seek to solve for Drupal?
- What has changed since version 2.0?
- What is a services server?
- How does authentication work in services?
- What is a resource in service?
- How does it all fit together?
- Testing and services.

What is the services module?

- Allow third party sites to communicate with Drupal using multiple protocols.
- Allow sites to expose data and allow updates
- D.O stats for usage
- What comes with core services
 - Servers
 - Resources
 - Authentication
- Like the rest of Drupal it is extensible other contrib modules leverage services.

Changes since 2.0

- Switch from the concept of services to that of resources
 - Changes started in patch committed in Paris
 - Bolt on to a system effectively designed for flash style communications
 - Attempt to modernise services for drupal and make the system more flexible in terms of how it could be exposed to the outside world
- Utilisation of authentication methods that are standards rather than keyauth - willing to embrace concepts created else where.
- Decision taken that services should be an enabling framework rather having a resource for 'everything'.
- Removed Browser
- Services 2.0 is no longer supported.

New features 3.0

- Pluggable auth
 - Now fully integrated and designed to allow for server and resource level controls
- Rest server in core
 - Allows most clients to communicate with services without the need for additional modules
- Response formats
 - Got rid of the need for JSON-Server and lets you define any type of return type based on ACCEPT headers
- Endpoints
 - Provides additional flexibility for configuration

Key concepts in services 3.x

- What modules services has dependencies upon
 - Ctools - exportables
 - Libraries - third party libraries
- What is a server?
 - Endpoints
- What is a resource?
 - CRUD
 - Actions Targeted Actions Relationships
- What is an authentication method?
 - The importance of security
 - Is it secure - drupal security team
 - Leverages core login mechanisms or OAuth

Servers in services

- The key building block for communicating.
- Decide the protocol you want based on how you want to respond.
- How to define a new server
 - How does a service integrate with core?
 - A walk through of the hooks in REST server
 - hook_server_info
 - hook_server
- Request and response formats
 - XMLRPC it is simple
 - REST many more options....but you have control
- First building block in an endpoint provides a way to communicate with the outside world.

Authentication in services

- Authentication protects your site
 - Always expose the least number of resources via your server endpoints
 - Always ensure you use an authentication method unless you have a good reason not too.
- The concept of sandboxed authentication..you are anonymous until proven otherwise.
 - Why is this important?
- Session authentication
 - Simplest authentication method.
 - Based on using drupal users directly
 - Login resource provides session information
 - All following requests provide session as header cookie

Authentication in services cont

- What is Oauth?
- How does it work?
- 2 legged vs 3 legged Oauth
- Spritual sucesor to key auth
- How to utilize the authenication plugin mechanism
 - No its not ctools!
 - What do the hooks do?
 - `hook_services_authentication_info`
 - Simple example of custom auth method

Resources in services

- Resources are the key concept in services
 - They actually do things everything else is just framework!
- How do we decide what resources are in services core?
 - Don't worry you can make your own
- Types of resources
 - CRUD
 - Index
 - Action
 - Targetted action
 - Relationship
- A walkthrough of a resource defintion

Resources in services cont

- Autonomy of a resource
 - What do the parameters mean?
 - A quick walk through.
- Potential gotchas
 - No standard for return codes at the moment
 - Service definitions can not change before 4.x but we can add optional parameters
 - More on this later.....
 - Drupal core permissions vs authentication

Putting it all together

- The services runtime
 - `services.runtime.inc` - the engine of services
- So how is a request actually processed?
 - Endpoint/Server routing
 - Parameter validation and detection
 - Request authentication
 - Resource invocation
 - Resource response
 - Service response
 - What if there is a problem?
- The benefits of ctools
 - Going live

Everything changes

- Services is designed to be extensible
- There are many ways to alter things
- All sub systems servers, authentication and resources can be added too.
- Examples of other services resources that exist
 - Services view
 - menu service
 - search service

Tests

- We have good test coverage but it could be better
 - Especially for XMLRPC
- One of the original modules to sign up for contribute module bot testings
 - Currently have 1743 asserts
 - As per core new features need tests
- Can't emphasize too highly the benefits of automated testing
 - Have some issues as we depend on third party libraries
 - Always looking to increase tests and would love people to get involved.
- Provides confidence with complex issues especially in security related areas

Services 4.x

- It won't work with drupal 6.
- The rest is still ideas because of the wscii project
- Some thoughts
 - Remove all resources and make services core lean
 - Utilize ctools more
 - Inheritance for Services resources
 - Support sandboxing of API:
 - Add the ability to version resources this may make it into 3.x - <http://drupal.org/node/1436366>
- Let us know what you want and join us in the issue in shapping what 4.x becomes.
- Key issue currently is lack of documentation there will be services docs sprint on Friday come join us.

With thanks

- Number of other people who need thanks for getting services where it is today.
 - Greg Dunlap (Heyrocker) uber maintainer for services for many years
 - Hugo Wettenburg - the original architect of 3.x
 - Yuriy Gerasimov (ygerasimov) - simpletest
 - Pelle Wessman (Voxpelli) - Oauth
 - Scott Nelson (snelson) - the original creator of services
 - Christoph Otto (cotto) - D6 maintainer
 - And many more people who have contributed patches.
- We have our own IRC channel #drupal-services