Adding New Services in sipXconfig

sipXconfig used the RESTlet API to add RESTful services

> http://www.restlet.org

Please note, due to a bug in RESTlet, it requires HTTP PUT request must have a body. You can add space, or anything to the body to workaround it.

Service design

- Identify resouces and design the resource structure
- Determine which operations (GET, PUT, POST, DELETE) will be available for which resources
- Map resources into URI structure
- Determine representation for each resource (specific format XML/JSON)

Service implementation

- Implement resources
- Each resource is an instance of restlet *Resource* class. Every RESTful function is mapped into one or more Resource methods. For example *GET* method is implemented by *Resource.represent*
- Resource instances are managed by Spring. You can inject any *context* or *manager* that is necessary to perform service operations. For example *phonebook* service delegates bulk of the work to *PhonebookManager*.

- Write unit test for resource and representation
- Use mock objects to verify if injected *managers' are used in expected way (for example does _POST* really calls _save" method with expected parameters?).
- For resources verify XML/JSON representation generation and parsing.

- Implement representations
- sipXconfig services intended to be used pragmatically should at the minimum support XML and JSON representations. Read-only services marshal sipXconfig domain objects into representation.

- Write service system tests
- Those are external tests written in Java/Ruby/Python that excersise the service. They are similar to UI tests.