Full Stack Data Grid

stack.push(‘MongoDB’);
stack.push(‘ServiceStack’);
stack.push(‘dojo dgrid’);

Background and Format

- My Sources
  - Documentation/Source/Demos
  - StackOverflow
  - Github
- My Experience with
  - dgrid ~5 months*
  - ServiceStack ~ 6 months**
  - MongoDB ~ 12 months***

* A few hours here and there a week learning and tinkering
** Started with researching and tinkering a few hours a week and progressed to everyday use
*** Started utilizing it heavily for proof of concepts against MongoDB, just recently been catching back up and getting re-familiar with the capabilities.

stack.pop("dojo dgrid");

Conceptually

- A core set of modules for data grid implementations
- Building blocks that serve as extensions, plugins or mixins to the core modules
- Pick and choose the building blocks to meet an individual implementations needs in a "light-weight" manner
- Great extensibility across all modules for developers to provide custom behavior

stack.pop(“dojo dgrid”);

Static Grid

- Rows
- Columns
- Static data set

Core

- Grid

Extensions

- None

stack.pop("dojo dgrid");

Static Interactive Grid

- Rows
- Columns
- Different Selection Modes
- Keyboard Interaction
- Eventing

Core

- Grid

Extensions

- Selection
- Keyboard

stack.pop("dojo dgrid");

Interactive OnDemandGrid

- Formatting
- Styling
- Customizing
- Store backed

Core

- OnDemandGrid

Extensions

- Selection
- Keyboard

stack.pop("ServiceStack");

What I like

- Declarative Routing
- No fuss Json Serialization
- Services support different formats out of the box
- Overall Simplicity

More details on the subject by ServiceStack creator:
http://stackoverflow.com/questions/9699083/servicestack-vs-asp-net-web-api

stack.pop("ServiceStack");

Supporting the OnDemandGrid

- Request Object and Routing
- Service implementation
  - IService and Service
- Configuration
  - web.config httpHandler
  - AppHostBase

stack.pop(“MongoDB”);

The C# Driver

- Code driven schema
- Fluid development
- Simplicity in storing models