

# APIs and Modularization

Jason Grey

Group Technology Director, OLSON

# What is an API?

- Wikipedia says: a set [stuff] provided by libraries in order to support the building of applications.
- Everything people or applications can depend on.
- APIs provide the contract and zero or more implementations.
- The API is the public part that other developers end up using from your library. It typically hides complex implementation details.
- It's the semantic skin for your code libraries.

# Why define an API?

- Designing your code around APIs improves code quality.
  - Promotes separation of concerns.
  - Promotes reusability.
  - Promotes standardization.
- Makes it easier to parallelize the development of a large application.
- Helps to facilitate proper modularization.

# What makes a good API?

- Good naming conventions: makes it easy to learn, and self-documenting.
- The API should strongly encourage users of the code into appropriate usage patterns.
  - Pagination
  - Getting right-size datasets
  - Hiding complex implementation details.
- Provide for easy integration: announce events, provide extension points.

# How to design a good API

- Make sure your API is based on REAL needs: collect requirements/feedback from other developers.
- Concentrate on meaning, naming conventions, and existing standards.
- Don't expose more than what is needed. It's ok to add something later, but removing something will break applications.

# What to watch out for...

- Once people start using an API, it becomes something people rely on not to change, so think ahead a bit...
- When possible, try to think through (or better yet, actually code) at 3 implementations.

# Modularization

- Breaking up a system into smaller systems.
- In Flex Builder – rather than making one huge project, make a library (or many), and “assemble” them into an application.

# Benefits of modularization

- Faster compile times in Flex Builder in many cases.
- Allows systematic versioning & upgrades of libraries separate from one another.
- Makes it easier to reuse code. Creating an app becomes more of a Lego-block exercise.
- Allows for easy re-skinning, and repurposing of code. (Perhaps you want to do an online Flex application AND offline AIR application)

# How to modularize in Flex Builder

- Create an API library interfaces, value objects, events, and errors for a domain of your application. If you are programming and managing ALL of the implementations, they can go in here as well.
- Create one or more implementation libraries of that API based on who's coding them, and how you want to manage/version/release them.
- Create your application project, and set dependencies.

# Demonstration

- <http://code.google.com/p/swfcamp-2009-shopper>

# On the shoulders of giants...

- See Joshua Bloch's great presentation on API Design for a much more in-depth look:  
<http://lcsd05.cs.tamu.edu/slides/keynote.pdf>
- <http://openide.netbeans.org/tutorial/api-design.html>
- *Will Tracz - Confessions of a Used Program Salesman, Addison-Wesley, 1995*

# Q&A / Discussion

- Anyone have examples to share?
- Have an application you'd like to modularize?