Wirehose-Personalization

WireHose provides everything you need to create high-performance, scalable personalized applications. It models users, guest users, channels, global channels, and channel factories, and includes optimized implementations which handle caching and propagation of default settings to individual sessions.

Users

WHUser is the parent entity for all WireHose users. It includes many methods for fetching, sorting and filtering a user's channels. Subclasses can override the

```
allUserChannels
```

method to include channels which are not modeled as part of the base "channels" relationship. Users can be automatically logged in via cookies, and you can override the default authentication and guest user creation behavior by implementing **WHApplicationHelper**'s delegate methods.

Channels

WHChannel defines an interface for things which can be personalized for individual users. WireHose includes Xcode templates which use **WHConcreteChannel** as a default implementation, or you can create your own channel objects by implementing the WHChannel interface. WireHose also provides support for global channels, which belong to all users.

Channel factories

WHChannelFactory defines an interface for objects which create channels for individual users and global channels. Channel factories also may provide default settings which can be overridden on a per-instance basis by individual channels. In a typical deployment scenario, channel factories are shared between all sessions in an application, and thus are often used to cache the results of expensive fetches. It includes a static inner class which provides a default implementation of the interface. WireHose includes Project Builder templates so new channel factories can be rapidly built.

Fetchers

WHFetcher defines an interface for objects which cache and filter the results of expensive fetches. **WHConcreteFetcher** provides an abstract implementation of this interface as a channel. For maximum performance, fetcher channels will attempt to use their factories' fetch results rather than perform their own whenever possible. **WHTagFetcher** is a fetcher which uses a **WHTagDataSource** to fetch taggable and indexable objects. **WHQualifierFetcher** uses a **WHQualifierDataSource** to retrieve enterprise objects by arbitrary qualifiers, making it well-suited for creating enterprise information channels that personalize queries into non-WireHose databases.