

Method Names for Data Retrieval

Posted At : June 19, 2009 7:15 AM | Posted By : Steve

Related Categories: ColdFusion

Ben Nadel recently posted a blog entry about [moving queries to service layers](#) (something I am a big fan of). Both the blog entry and the comments that followed were great and well worth the read. I especially liked Hal Helm's comment about differentiating between queries that are necessarily the same and those that are accidentally the same.

One thing that came up in this discussion was how to name the methods that retrieve data. One example was "getArticlesByAuthorID()" (I'm just picking one of many that followed this pattern). To me, this is a code smell. The method name describes not only what you want, but how you are going to get it.

I would do it differently.

If I am writing a page and I need to get articles for a certain author and I don't already have any methods written to get this kind of information, I would create a method name "getArticles" because "Articles" are what I want to get. Then I would have "AuthorID" as a required argument.

So, the code for my method would start like this (including only relevant attributes):

```
<cffunction name="getArticles" access="public">
    <cfargument name="AuthorID" required="true">

    ...

</cffunction>
```

Now my method name describes what I want, not how I get it. It is obvious from the required argument, however, how I am going to get the articles.

Then I would write whatever code needed to get articles based on the AuthorID.

This would solve the immediate problem, but it would start a nice API to build on in the future.

Now, if I later discover that I want to get the same basic information except that I want to get it by CategoryID, I can do the following:

```
<cffunction name="getArticles" access="public">
    <cfargument name="AuthorID" required="false">
    <cfargument name="CategoryID" required="false">

    <cfif NOT ( StructKeyExists(arguments,"AuthorID") OR StructKeyExists(arguments,"CategoryID") )>
        <cfthrow message="getArticles requires either AuthorID or CategoryID">
    </cfif>

    ...

</cffunction>
```

For me the code inside the method might be as simple as passing the arguments to DataMgr.getRecords(), but it also might be a SQL query with some conditionals in the WHERE clause. I might even discover the need to make a private getArticlesByAuthorID method and a private getArticlesByCategoryID method.

The point is, however, that the code needed to retrieve the data shouldn't impact the API. The API should describe what you want, and not be impacted by what steps it will take to get it.