Article
Istvan Hahn · Dec 20, 2016 5m read

Atelier, The Productivity Boost Tool

Development productivity? Isn ’ t it kind of an oxymoron? Anyhow, we all dream a magic tool which does the tedious job of the software development and leave the intuitive or creative part on the developer. Atelier is the next generation tool of InterSystems aiming to boost the productivity of Caché/ Ensemble based software development. Atelier is based on the popular Eclipse IDE platform, which makes available the “ infinite ” pool of third party tools for Caché developers. Those tools can truly boost the productivity. This short article gives some examples how third party plug-ins can help developing for Caché. It is an absolute beginner ’ s guide, beginner to Eclipse but experienced to Caché.

The first step

Eclipse is an Integrated Development Environment (IDE). A “ smart text editor ” . Like emacs for UNIX, EDT for RSX or TPU for VMS, Eclipse provides not just the text editing feature but also gives a programming interface to enhance those features. This extensibility makes IDEs being able to facilitate complex development projects where the solution is built of various technologies using several programming/ configuration languages. Programming extensions of Eclipse are called “ plug-in ” . A plug-in can be plugged in via two distinct methods.

The classic method is to download a plug-in from a repository and install. You can do it from the “ Help -> Install New Software ” menu of Atelier. That is clear.

The other option is to install from Eclipse Marketplace. In order to take products from a marketplace you need install the “ Marketplace Client ” plug-in using the classic method. On completing the install you are going to see a new menu item in the Atelier Help menu: “ Eclipse Marketplace ” .

Ready? Before you go, please note!

Plug-in install changes the way Atelier works. Please make sure that a particular plug-in does not compromise the expected functionality of Atelier prior installing it!

CSP Development

Although developing web application with CSP is not the promoted way anymore, there are numbers of applications are still based on it. CSP using both static content (such as CSS, JS) and active HTML (CSP, CSR, CLS). In particular CSS uses lots of keywords hard to remember and yet easy to misspell. Just as an example:

What is correct in CSS to hide a text?

1. *Visible: 0*

2. *Display: none*

3. Hidden: true

4. Opacity: 1

Would not it be nice to have code completion for CSS?

There are plenty of HTML, CSS, JS editor plug-in available for Eclipse. My personal favorite is Eclipse Web Development Toolkit, which is a classic plug-in. It does pretty much you need for editing static web content. Since you are plugging in WDT into Atelier, it is obvious that the server synchronization works as expected. Only the source editor is replaced.

Static WEB content

The content editor for CSS, JS, HTML, etc. is replaced. Everything works perfect, the server integration is smooth.

Boost factors:

n Code completion for standard CSS

n Code completion for standard HTML

n IntelliSense for Java script

Active HTML content

CSP falls to the " Active Server Pages " category of the web development technologies. Such technologies are using HTML as a base language to create a page, but the source holds instructions to a server side text processor. Those instructions are swallowed by the processor, so the page sent to the browser does not contain them. Instructions are very much dependent on the actual technology. It is nothing different with CSP. In other words: a CSP file is a combination of HTML and server side script. Obviously Eclipse does not provide yet an off-the-box solution for Caché Object Script. Therefor I advise to take special attention to active HTML content. When the server side script makes the dominant part of a source, than Atelier HTML editor is efficient. When it is more about native HTML and client side script, than Eclipse HTML editor is more productive. The good news is that you do not need to make an exclusive decision. You can use both even at the same time. Just right click on the file you are about to edit, and select the desired editor from the " Open with " menu item. The default editor can be set at " Preferences -> General -> Content Type -> Text -> HTML " by adding the " *.csp " to the list.

There is one issue here. When the Eclipse HTML editor is used, the compile button does not synchronize to the server. However the save button does both save and compile.

CSR? I strongly recommend to use Atelier macro editor. That is the dominant language, not HTML.

Angular

Are you building content rich web application? Are you using Angular at client? I have a good news to you. You can obtain AngularJS or Angular2 plug-in from Eclipse marketplace. It is built on top of Eclipse Web Development Toolkit Platform (WTP). Which is already installed at your computer if you followed the earlier section.

XML Schema editor

My personal favorite is the " Eclipse XML Editors and Tools " classic plug-in. I can not imagine any more time-consuming task than XML Schema editing or XSLT validation. Thanks to the authors of the plug-in I (and you too) can happily rescue. It has a graphical design tool for XML Schema creation.

SQL Development Tools

The name of the plug-in is “ SQL Development Tools ” . And what it provides (surprisingly) SQL development tools. It does all kinds of SQL statements, not just query. Try this guy and you would never ever want anything else. It is an Eclipse marketplace plug-in. Works perfectly with Caché. The SQL editor has code completion and offers the table names. You can use graphical query builder too. Here in the query builder the column names are also available even in the text editor.

The first time setup takes a little while, but it is because there is no Caché SQL dialect shipped with the plug-in. During the setup just say Caché is a Generic JDBC data source. You must provide the JDBC driver location and name, and the REST is just straightforward. Once Eclipse connects to the server, you will feel like you were using it since decades… Obviously because there is no Caché dialect yet, none of the Caché SQL extension works or at least offered by the IntelliSense.

RESTful API Documentation

The tool called repreZEN, available from the Eclipse Market Place. It is a commercial application. What it is useful for is designing/ documenting a RESTful API. It provides a nice yaml editor. The edited model is in Swagger format with IntelliSense. The model also could be viewed in a graphic viewer. The feature set contains code generation. Unfortunately not for Caché (yet). Even though I learnt a lot from the API design/ documentation. Because repreZEN is not designed for Atelier, installing it has several side effects. One is very annoying: Atelier never stops complaining that the files used are not synchronized to server. The other is just inconvenient: repreZEN creates its own perspective which is set as default during the installation. That is you have to set the default perspective back to Atelier manually. I mean if you want.

Epilogue

I know leaving the good old tool (Caché Studio) behind and jump into something new (Atelier) is sometimes painful. I hope this article gave you enough courage to start working with Atelier. Please keep always in mind what you can win by using plug-ins.

Happy plugin browsing…

#Beginner #Caché

Source URL:https://community.intersystems.com/post/atelier-productivity-boost-tool