

Avista Template Engine

Abstract

Kiev Software Factory Ltd. announces commercial availability of its Avista Template Engine (ATE). This fast, robust and feature-full template engine is specially designed for web development teams to achieve the highest degree of concern separation and high productivity rate, and to decrease the cost of quality assurance. The main advantage of using ATE is to achieve a workflow optimization between developers and web designers, allow independent and simultaneous user interface (UI) design and behaviour implementation, enable customers to review and approve UI and content at the early stages of development, facilitate reusability, internationalization, automatic regression testing, and provide with even bigger set of beneficial features as described below.

Introduction

Facing numerous and challenging list of requirements while setting off a new project, Kiev Software Factory had carried out extensive research into web application development techniques and strategies, studied and examined several mainstream web frameworks and template engines as well as frameworks using fairly rare approaches to web development, and came up with the number of design decisions that were later embodied in the Avista Template Engine.

The ATE has become the core tool in one of KSF's projects and evolved into a proven efficient tool. The ATE has been successfully used by KSF for developing presentation layer of several enterprise scale applications. It has also been involved into the integration with third-party products, and has been integrated with CruiseControl to create fully automatic functionality and regression testing facilities.

Goals

At first, the Avista Template Engine was designed and implemented with following goals in mind:

- Suit for developing server-side applications with complex UI
- Allow switching UI appearance instantly without changing any code
- Provide explicit, simple and easy-to-learn programming interface
- Support internationalization and localization

As it turned out, the ATE allowed developers to:

- Address efficiency, quality, reliability, stability
- Assist development as well as quality assurance and project management
- See features beyond...

Features overview

Concern separation

One of the most outstanding features of the ATE is that it does enforce concern separation between template designers and application developers. For instance, there is a mechanism for simple switching of web page appearance to "printer-friendly" or "mobile-device-friendly" look – a designer only has to provide a new template while a developer is not concerned at all.

Testability improved

When it comes to testing, few available frameworks are able to provide developers and QA engineers with an acceptable set of tools to control over functional and regression problems. Developers using the ATE can easily separate business data from the user interface mark-up to enable automation of the functional regression testing. A QA engineer needs to perform a test case manually only once to make the framework record a test. Since the test has been recorded it will be run during a project build to detect any regression that might occur.

Aid to project elaboration and approval

The ATE can be used for prototyping. The template engine allows to produce accurate user interface mock-ups, which can be filled with real or test data, so that a customer can approve user interface appearance and contents as soon as possible – at the software design stage.

Arbitrary order of content generation

The ATE allows building web pages in any arbitrary order that is suitable for the implementation of the application instead of the order that is imposed by technical constraints intrinsic to technologies such as Java Server Pages.

Ability to generate any text content

Explicit and simple mechanism of producing text content can be used for generating any text content such as plain text, XML, HTML, CSV, etc. Furthermore, flexible syntax of the ATE language even allows XML targeted templates stay a well-formed and valid XML documents. Once developers have learned the ATE for HTML generation they don't have to learn another library for generating XML.

Easy to learn, straightforward and simple API

No other template engine can compete in simplicity with ATE, comprising of only two basic interfaces. A software library has never been that simple!

ATE based applications are easy to debug

Additional pseudo-templates are available to facilitate debugging

AJAX-enabled

Employ modern AJAX technique of web development using ATE as it supports AJAX transparently and allows Java developers to give up thinking of the AJAX-specific coding and concentrate on the business logic. Code written once will work for both AJAX-enabled and regular templates.

Extendable

The ATE allows developers to expand engine capabilities with little efforts. Any generation rules can be applied by merely extending classes of the ATE.

Localization and internationalization support

Localization and internationalization have been among primary aims of the ATE design and implementation and supported naturally.

Template tiles

The ATE supports so called 'tiles' that are simply reusable building fragments that can be assembled in compile time into the final template.

Adobe® Dreamweaver® support

Currently Adobe® Dreamweaver® is supported so that it can process template tiles.

Notes

Adobe® Dreamweaver® is a registered trade mark of Adobe Systems, Inc.
Java® is a registered trade mark of Sun Microsystems, Inc.