

Google App Engine Java и Google Web Toolkit: разработка Web-приложений

РАЗРАБОТКА
Web-ПРИЛОЖЕНИЙ
ДЛЯ Google App Engine

ИСПОЛЬЗОВАНИЕ
ПРОГРАММНОГО ИНТЕРФЕЙСА
СЛУЖБ Google App Engine

СОЗДАНИЕ Ajax-ПРИЛОЖЕНИЙ
НА ОСНОВЕ Google Web Toolkit

КОМПОНЕНТЫ
GUI-ИНТЕРФЕЙСА
Google Web Toolkit

РАБОТА GWT-ПРИЛОЖЕНИЯ
НА СТОРОНЕ СЕРВЕРА



PRO

ПРОФЕССИОНАЛЬНОЕ
ПРОГРАММИРОВАНИЕ

УДК 004.43:004.738.5
ББК 32.973.26-018.2
М38

Машнин Т. С.

М38 Google App Engine Java и Google Web Toolkit: разработка Web-приложений. — СПб.: БХВ-Петербург, 2014. — 352 с.: ил. — (Профессиональное программирование)

ISBN 978-5-9775-0828-5

Книга посвящена разработке Web-приложений для платформы Google App Engine и на основе фреймворка Google Web Toolkit на языке программирования Java и с использованием среды разработки Eclipse. Рассмотрено создание проектов и запуск GWT-приложений и приложений для Google App Engine. Описано использование программного интерфейса служб платформы Google App Engine, создание GUI-интерфейса на основе фреймворка Google Web Toolkit, оптимизация и интернационализация GWT-приложения. Показано применение фреймворков UiBinder и Activities and Places для разработки клиентской части GWT-приложения, а также фреймворков GWT RPC и RequestFactory для разработки серверной части GWT-приложения.

Материал книги сопровождается большим количеством примеров с подробным анализом исходных кодов.

Для программистов

УДК 004.43:004.738.5
ББК 32.973.26-018.2

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Игорь Шишигин</i>
Зав. редакцией	<i>Екатерина Капалыгина</i>
Редактор	<i>Анна Кузьмина</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Корректор	<i>Зинаида Дмитриева</i>
Дизайн серии	<i>Инны Тачиной</i>
Оформление обложки	<i>Марины Дамбиевой</i>

Подписано в печать 02.08.13.
Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 28,38.
Тираж 1000 экз. Заказ №
"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20.

Первая Академическая типография "Наука"
199034, Санкт-Петербург, 9 линия, 12/28

ISBN 978-5-9775-0828-5

© Машнин Т. С., 2014
© Оформление, издательство "БХВ-Петербург", 2014

Оглавление

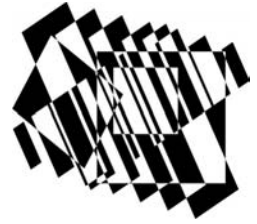
Введение	7
Платформа Google App Engine	7
Фреймворк Google Web Toolkit	9
ЧАСТЬ I. ПЛАТФОРМА GOOGLE APP ENGINE	11
Глава 1. Начало работы с Google App Engine	13
Установка инструментов разработки	13
Создание проекта приложения	18
Запуск приложения из среды Eclipse	23
Развертывание приложения на платформе App Engine	25
Регистрация приложения	25
Страница администрирования приложением	26
Загрузка приложения в App Engine	31
Создание пользовательских разделов консоли администрирования	32
Глава 2. Журналирование приложения	34
Библиотека <i>java.util.logging</i>	34
Библиотека Log4j	37
LogService API	38
Глава 3. Определение местоположения пользователя	42
Глава 4. Аутентификация пользователей	45
Ограничения доступа к ресурсам в дескрипторе web.xml	45
Программный интерфейс Users API	48
Аутентификация с помощью Google Accounts	49
Аутентификация с помощью OpenID	50
Глава 5. Использование JSP/JSF-страниц в GAE-приложении	54
Технология JSP	54
Использование JSTL	57
Технология JSF	60

Глава 6. Хранение данных приложения	65
App Engine Datastore.....	65
Datastore API	67
Служба Remote API	80
JDO и JPA.....	82
JDO.....	82
JPA	91
Objectify	100
Twig.....	108
Slim3.....	116
Google Cloud SQL.....	123
Google Cloud Storage и Blobstore.....	126
Google Cloud Storage.....	126
Blobstore.....	128
Сервис изображений.....	132
Служба Memcache.....	134
Глава 7. Поддержка сессий и HTTPS	137
Поддержка протокола HTTP/SSL.....	137
Использование сессий и cookie	138
Глава 8. Сервисы сообщений Mail, XMPP и Channel	140
Служба Mail	140
Отправка сообщений электронной почты	140
Получение сообщений электронной почты.....	141
Пример использования службы Mail	143
Служба XMPP	146
Отправка мгновенных сообщений.....	147
Получение мгновенных сообщений.....	147
Пример использования службы XMPP	150
Служба Channel.....	153
Глава 9. Фильтры и обработка ошибок.....	159
Фильтрация запросов и ответов	159
Обработка ошибок.....	161
Глава 10. Разработка Backend-приложений	164
Глава 11. Использование протокола OAuth 2.0 для получения доступа к Google-сервисам	168
Служба URL Fetch	172
Глава 12. Запланированные задачи и очередь задач	175
Служба Cron.....	175
Служба Task Queue.....	177
Очереди Push.....	177
Отложенные задачи <i>DeferredTask</i>	180
Очереди Pull	181

Глава 13. Службы поиска Search и Prospective Search	183
Служба Search	183
Служба Prospective Search.....	187
ЧАСТЬ II. ФРЕЙМВОРК GOOGLE WEB TOOLKIT	193
Глава 14. Начало работы с Google Web Toolkit.....	195
Установка плагинов фреймворка GWT	196
Создание проекта GWT-приложения.....	197
Структура проекта GWT-приложения	197
GWT-модули	198
Конфигурационный XML-файл определения GWT-модуля.....	200
Модель программирования фреймворка GWT	203
Запуск GWT-приложения в режиме разработки.....	208
Запуск GWT-приложения как Web-приложения	210
Глава 15. Компоненты графического интерфейса пользователя.....	214
Кнопка <i>Button</i>	215
Отличие метода <i>setText()</i> от метода <i>setHTML()</i>	215
Обработчики событий кнопки	216
Определение свойств кнопки.....	222
Кнопка <i>PushButton</i>	222
Переключатель <i>RadioButton</i>	224
Флажок <i>CheckBox</i>	225
Компонент выбора даты <i>DatePicker</i>	226
Кнопка <i>ToggleButton</i>	229
Текстовое поле <i>TextBox</i>	231
Поле ввода пароля <i>PasswordTextBox</i>	232
Текстовая область <i>TextArea</i>	233
Гиперссылка <i>Hyperlink</i>	234
Гиперссылка <i>Anchor</i>	236
Список выбора <i>ListBox</i>	237
Компоненты <i>Cell Widgets</i>	238
Столбец <i>CellList</i>	239
Таблица <i>CellTable</i>	249
Таблица <i>DataGrid</i>	255
Дерево <i>CellTree</i>	257
Дерево <i>CellBrowser</i>	259
Панель меню <i>MenuBar</i>	260
Дерево <i>Tree</i>	263
Поле подсказки <i>SuggestBox</i>	265
Редактор текста <i>RichTextArea</i>	267
Таблица <i>FlexTable</i>	269
Таблица <i>Grid</i>	271
Всплывающие окна <i>PopupPanel</i> , <i>DecoratedPopupPanel</i> , <i>LoggingPopup</i> и <i>DialogBox</i>	273
Уведомление <i>NotificationMole</i>	276
Панели с закладками <i>TabPanel</i> и <i>TabLayoutPanel</i>	276
Загрузчик файлов <i>FileUpload</i> и форма <i>FormPanel</i>	279
Скрытое поле <i>Hidden</i>	280

Фрейм <i>Frame</i>	281
Изображение <i>Image</i>	281
Метка <i>Label</i>	283
Метка <i>HTML</i>	284
Метка <i>InlineHTML</i>	284
Метка <i>InlineLabel</i>	284
Медиакомпоненты <i>Audio</i> и <i>Video</i>	284
Компонент <i>Video</i>	284
Компонент <i>Audio</i>	285
Компонент <i>Canvas</i>	286
Панели компоновки	287
Панели <i>AbsolutePanel</i> , <i>RootPanel</i> , <i>RootLayoutPanel</i> и <i>LayoutPanel</i>	287
Панели <i>StackPanel</i> , <i>DecoratedStackPanel</i> и <i>StackLayoutPanel</i>	290
Панели <i>DockPanel</i> , <i>DockLayoutPanel</i> и <i>SplitLayoutPanel</i>	292
Панели <i>HorizontalPanel</i> и <i>VerticalPanel</i>	294
Панель <i>FlowPanel</i>	295
Панель <i>HTMLPanel</i>	295
Панель <i>HeaderPanel</i>	296
Панели <i>SimplePanel</i> , <i>DecoratorPanel</i> , <i>FocusPanel</i> и <i>SimpleLayoutPanel</i>	296
Панель <i>ScrollPanel</i>	298
Панель <i>DisclosurePanel</i>	299
Панель <i>CaptionPanel</i>	299
Изменение внешнего вида GWT-компонентов	300
Фреймворк <i>UiBinder</i>	304
Глава 16. Интернационализация GWT-приложения	307
Статическая интернационализация	308
Динамическая интернационализация	312
Интернационализация <i>UiBinder</i>	312
Глава 17. Программный интерфейс JavaScript Native Interface	314
Глава 18. Оптимизация GWT-приложения	316
Разделение кода (Code Splitting)	316
Отложенное связывание (Deferred Binding)	318
Глава 19. Поддержка истории Web-браузера	321
Глава 20. Фреймворк Activities and Places	324
Глава 21. Взаимодействие GWT-приложения с сервером	330
Фреймворк GWT RPC	330
Библиотека GWT HTTP Client	335
Фреймворк RequestFactory	341
Литература	349
Предметный указатель	350

ГЛАВА 1



Начало работы с Google App Engine

Установка инструментов разработки

Разработка на основе платформы Google App Engine (GAE) подразумевает создание Web-приложений. Для разработки Web-приложений на языке Java платформа GAE предлагает использовать набор Eclipse-плагинов, облегчающих организацию проекта, тестирование и развертывание приложения. Поэтому в качестве первого шага скачаем по адресу <http://www.eclipse.org/downloads/> последнюю версию среды Eclipse IDE for Java EE Developers, обеспечивающую все необходимые инструменты разработки Java Web-приложений. При этом предполагается, что набор JDK платформы Java уже установлен на компьютере (<http://www.oracle.com/technetwork/java/javase/downloads/index.html>).

ПРИМЕЧАНИЕ

При написании книги использовался релиз Indigo SR2 среды Eclipse.

После распаковки скачанного дистрибутива запустим среду Eclipse, закроем страницу приветствия **Welcome** кнопкой **Workbench** и в меню **Help** выберем команду **Install New Software** — в результате откроется окно мастера инсталляции Eclipse-плагинов (рис. 1.1).

Нажмем кнопку **Add** и в поле **Name** введем **Google**, а в поле **Location** — адрес установки Google-модуля, для Eclipse Indigo <http://dl.google.com/eclipse/plugin/3.7> (рис. 1.2).

Нажмем кнопку **OK**. После поиска в Интернете в окне мастера отобразится набор плагинов Google-модуля (рис. 1.3).

Плагин Google App Engine Tools for Android обеспечивает создание клиент-серверных приложений для платформы Android, в которых клиентская часть размещается в мобильном Android-устройстве, а серверная часть — в облаке GAE-платформы. Данный плагин требует предварительной установки плагина Android Development Tools (ADT) для разработки Android-приложений в среде Eclipse.

Плагин Google Plugin for Eclipse обеспечивает поддержку проектов GAE- и GWT-приложений в среде Eclipse.

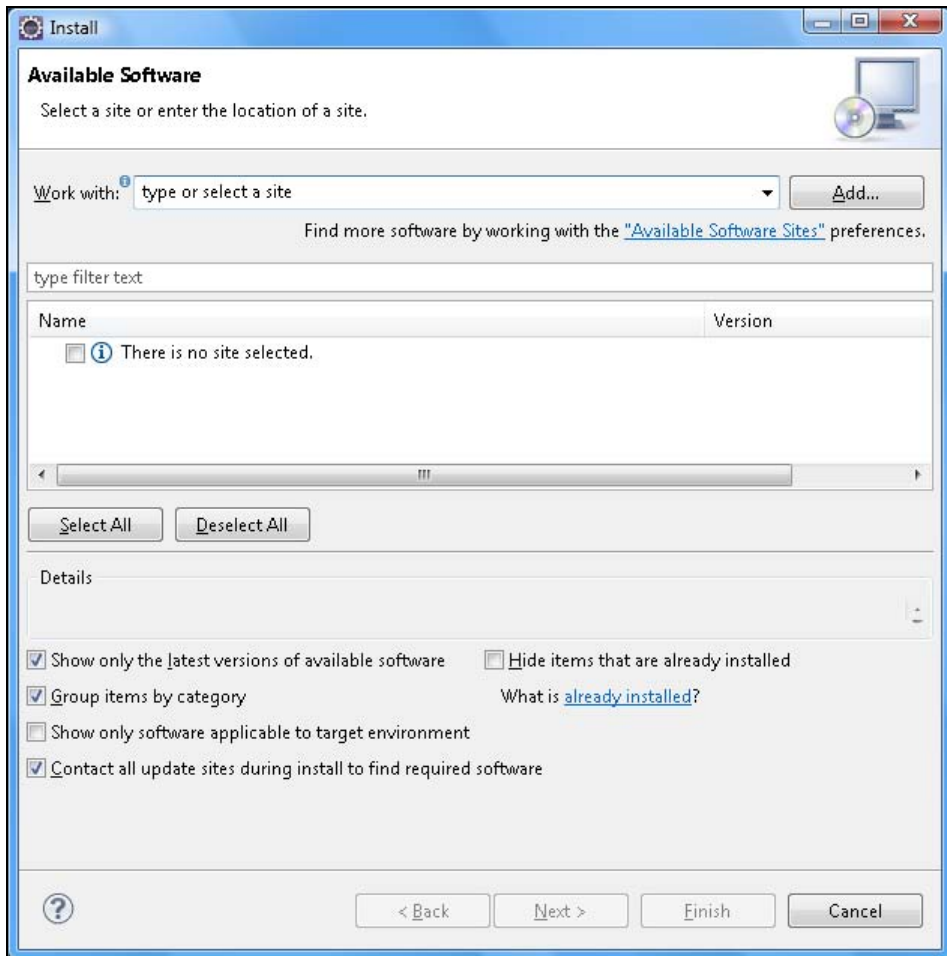


Рис. 1.1. Мастер установки Eclipse-плагинов

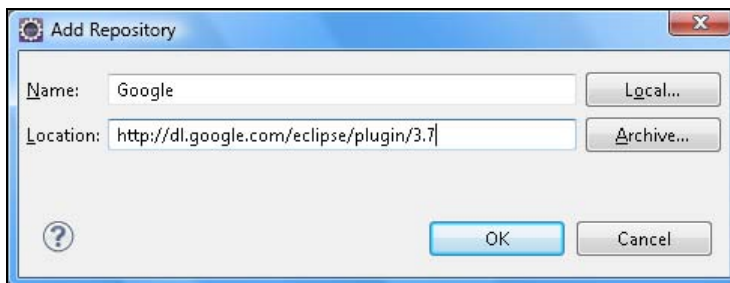


Рис. 1.2. Определение адреса установки Google-модуля

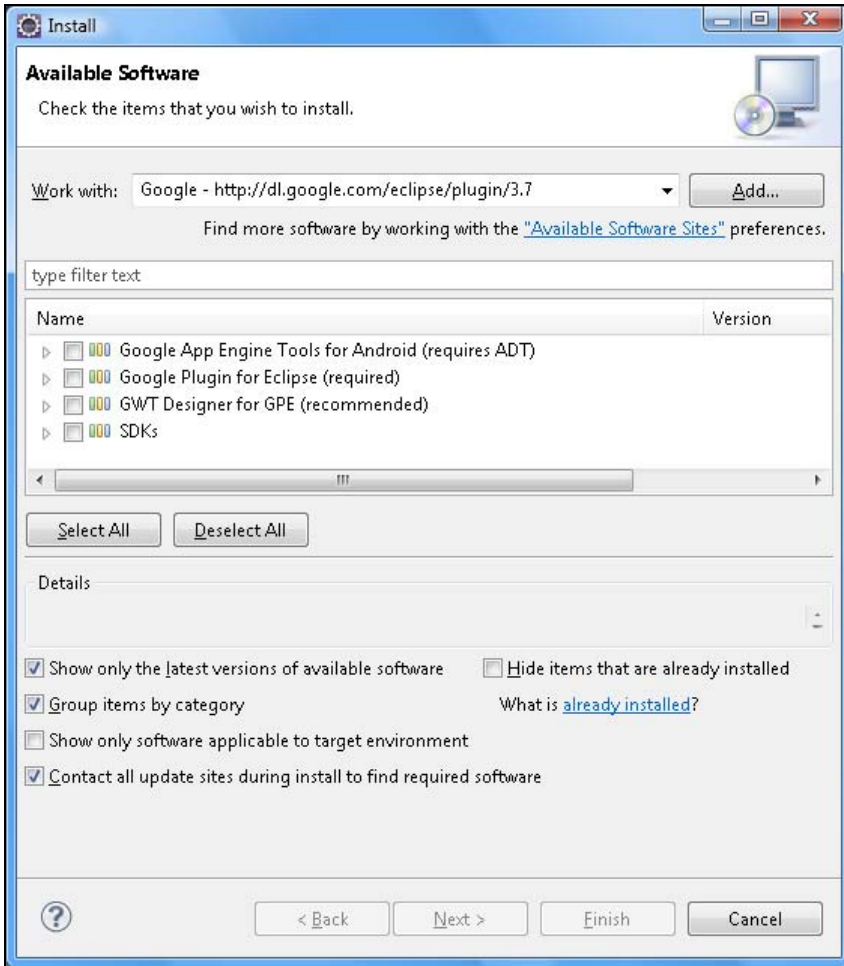




Рис. 1.3. Набор плагинов Google-модуля для среды Eclipse

Плагин GWT Designer for GPE предоставляет средства визуального редактирования GUI-интерфейса GWT-приложений.

Плагин SDKs обеспечивает установку наборов Google App Engine Java SDK и GWT SDK, необходимых для разработки GAE- и GWT-приложений в среде Eclipse.

Для начала отметим флажки **Google Plugin for Eclipse** и **Google App Engine Java SDK**, нажмем кнопку **Next** мастера и, следуя инструкциям, установим выбранные плагины с перезапуском среды Eclipse.

В результате установки плагина Google Plugin for Eclipse (GPE) в Workbench-окне среды Eclipse появятся кнопки **Google Services and Development Tools**  и , а в диалоговом окне команды **New | Other** меню **File** появятся разделы **Google** и **Google Web Toolkit** (рис. 1.4).

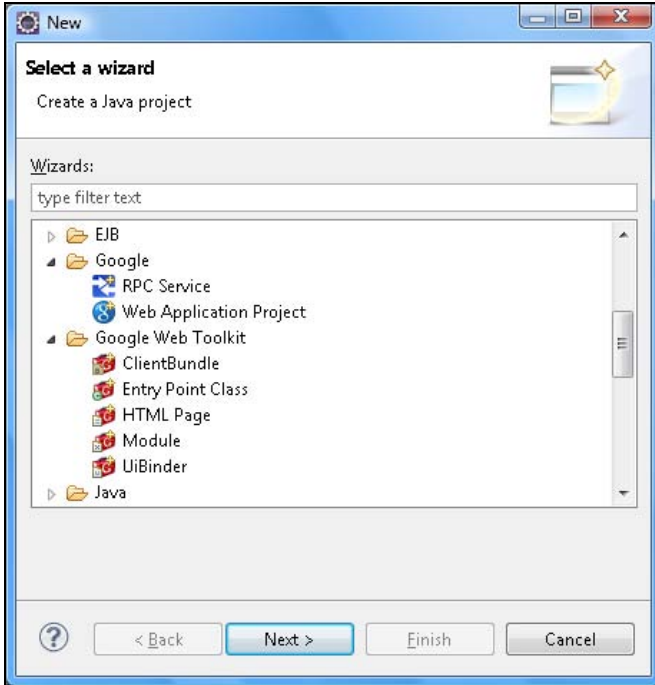


Рис. 1.4. Разделы Google и Google Web Toolkit окна команды New | Other меню File

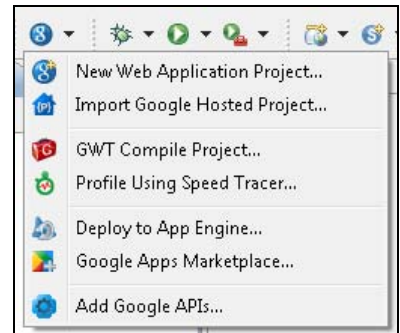


Рис. 1.5. Команды кнопки Google Services and Development Tools

Кнопка **Sign in to Google** позволяет пройти аутентификацию/авторизацию для доступа к таким Google-сервисам из среды Eclipse, как Project Hosting, Google SQL Service, Google Apps Marketplace и Google App Engine.

Сервис Project Hosting (<http://code.google.com/hosting/>) предоставляет возможность размещения открытых проектов в удаленном VCS-репозитории (Version Control System) для организации командной работы над ними. Данный сервис поддерживает такие системы контроля версий, как Subversion, Mercurial и Git.

Сервис Google SQL Service представляет собой Web-сервис, обеспечивающий создание, администрирование и использование MySQL базы данных для GAE-приложения.

Сервис Google Apps Marketplace (<http://www.google.com/enterprise/marketplace/>) является магазином Web-приложений, которые интегрируются с бизнес-приложением Google Apps (<http://www.google.com/enterprise/apps/business/>).

Раскрытая кнопка **Google Services and Development Tools** показывает набор команд (рис. 1.5):

- ◆ **New Web Application Project** — запускает мастер создания проекта приложения платформ GAE и GWT;
- ◆ **Import Google Hosted Project** — позволяет импортировать проект, размещенный на хостинге сервиса Project Hosting, в Workspace-пространство среды Eclipse;

- ◆ **GWT Compile Project** — запускает мастер компиляции Java-кода проекта GWT-приложения в JavaScript-код;
- ◆ **Profile Using Speed Tracer** — обеспечивает анализ производительности приложения с помощью инструмента Speed Tracer — расширения Web-браузера Google Chrome;
- ◆ **Deploy to App Engine** — запускает мастер развертывания приложения на GAE-платформе;
- ◆ **Google Apps Marketplace** — обеспечивает загрузку приложения в магазин Google Apps Marketplace;
- ◆ **Add Google APIs** — позволяет добавить в путь приложения программный интерфейс API различных Google-сервисов.

Раздел **Google** команды **New | Other** меню **File** среды Eclipse содержит мастера:

- ◆ **RPC Service** — на основе Entity-компонентов создает для GWT-приложения RequestFactory-сервис данных;
- ◆ **Web Application Project** — создает проекты GAE- и GWT-приложений.

Раздел **Google Web Toolkit** команды **New | Other** меню **File** среды Eclipse содержит мастера:

- ◆ **ClientBundle** — создает для GWT-приложения ClientBundle-интерфейс, группирующий ресурсы приложения, такие как изображения, CSS-стили, текстовые файлы и др., в единый модуль, что ускоряет загрузку и запуск клиентской части приложения;
- ◆ **Entry Point Class** — создает для GWT-приложения EntryPoint-класс — точку входа в GWT-модуль;
- ◆ **HTML Page** — создает для GWT-приложения HTML-страницу, обеспечивающую загрузку клиентской части приложения;
- ◆ **Module** — создает для GWT-приложения новый GWT-модуль;
- ◆ **UiBinder** — создает для GWT-приложения UiBinder-шаблон, декларативно определяющий GUI-интерфейс приложения.

Для использования в GAE-проекте страниц Java Server Pages (JSP) требуется Java-компилятор `javac.exe`, содержащийся в наборе Java SE Development Kit (JDK). По умолчанию среда Eclipse использует не набор JDK, а среду выполнения JRE, поэтому после установки плагинов GPE и GAE Java SDK, перед созданием GAE-проекта, в меню **Windows** среды Eclipse выберем команду **Preferences**, в диалоговом окне которой откроем раздел **Java | Installed JREs** и кнопкой **Add** добавим предварительно инсталлированный набор JDK (<http://www.oracle.com/technetwork/java/javase/downloads/index.html>) (рис. 1.6).

В настоящее время GAE-платформа поддерживает версии Java 5 и 6, поэтому необходимо проверить, чтобы уровень компиляции, установленный в разделе **Java | Compiler** диалогового окна команды **Preferences**, а также версия Java раздела **Project Facets** диалогового окна свойств GAE-проекта соответствовали версии Java раздела **Java | Installed JREs**.

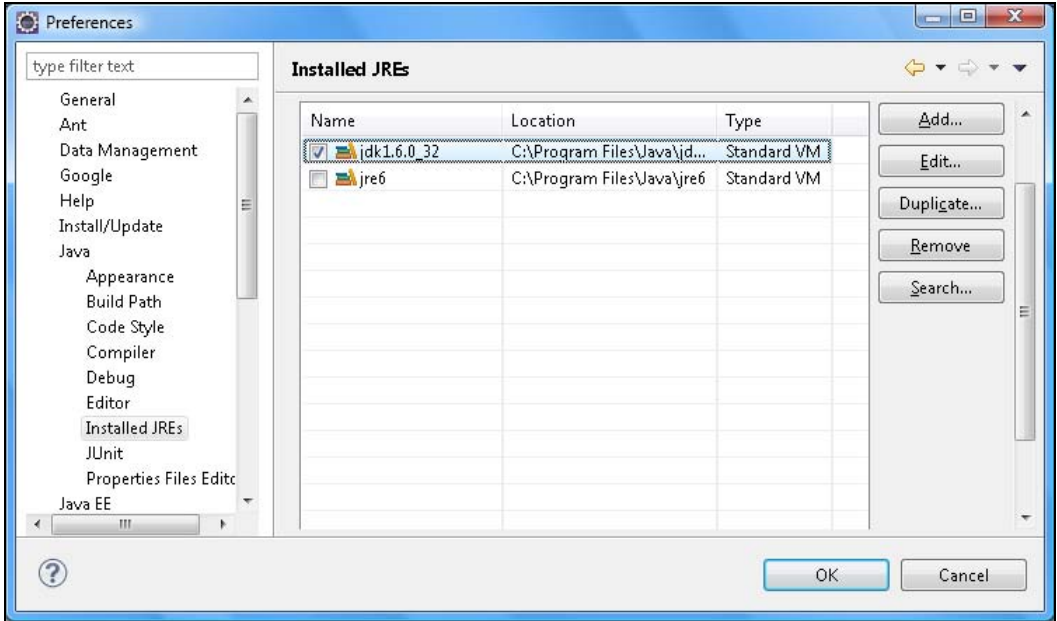


Рис. 1.6. Установка использования JDK по умолчанию

Создание проекта приложения

Теперь все готово для создания проекта GAE-приложения. В меню **File** среды Eclipse выберем команду **New | Other | Google | Web Application Project** и нажмем кнопку **Next**. В окне мастера создания проекта, в поле **Project name** введем имя проекта, в поле **Package** — имя пакета проекта, отметим флажок **Use Google App Engine** и нажмем кнопку **Finish** (рис. 1.7).

В результате в Workspace-пространстве среды Eclipse будет сгенерирована основа проекта GAE-приложения.

В путь GAE-проекта будут добавлены библиотеки среды JRE и набора GAE Java SDK для локального запуска GAE-приложения, предоставляющего Servlet-контейнер Web-сервера Jetty и обеспечивающего эмуляцию работы GAE-служб.

Папка `src` сгенерированного GAE-проекта содержит пакет с классом сервлета, который расширяет класс `javax.servlet.http.HttpServlet` и отвечает за обработку HTTP-запросов к приложению (листинг 1.1). Сгенерированный Servlet-класс переопределяет метод `doGet()` класса `HttpServlet`, обрабатывающий HTTP GET-запрос, в котором в ответ на запрос записывается строка "Hello, world". Для вызова данного сервлета его класс указан в дескрипторе `web.xml` развертывания приложения с помощью тега `<servlet-class>`, при этом тег `<url-pattern>` указывает относительный URL-адрес вызова сервлета. Дескриптор `web.xml` располагается в папке `war\WEB-INF` GAE-проекта. Тег `<welcome-file>` дескриптора `web.xml` определяет страницу приветствия приложения.

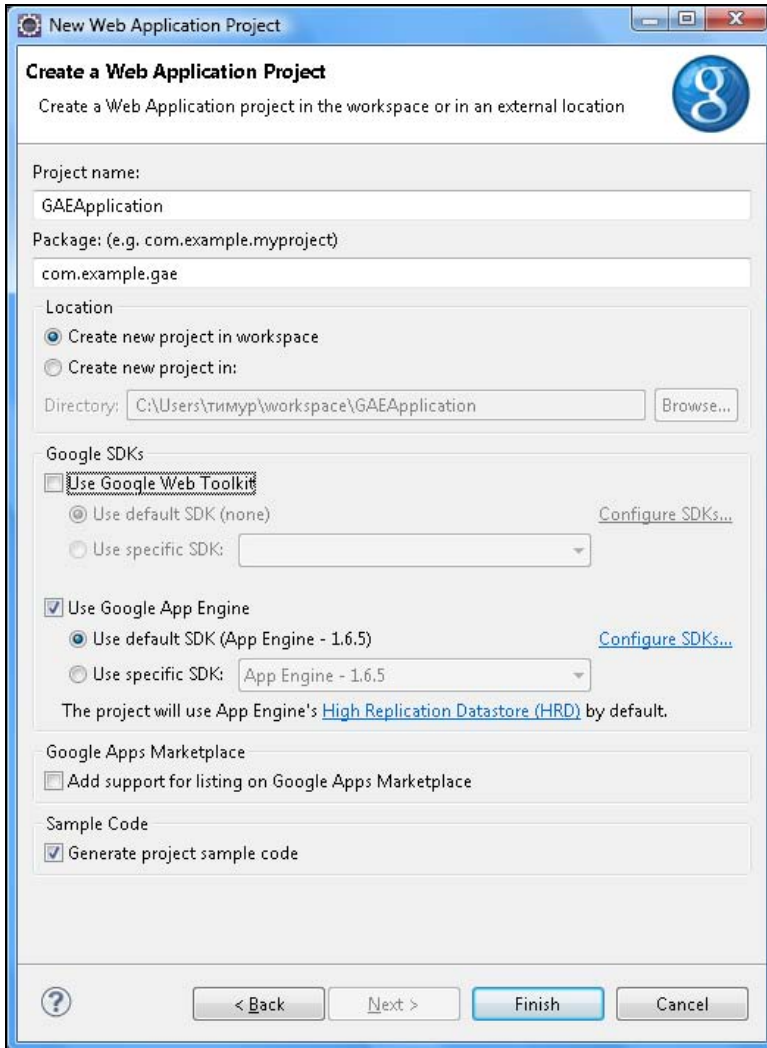


Рис. 1.7. GPE-мастер создания проекта

Листинг 1.1. Код Servlet-класса GAE-приложения

```

package com.example.gaeapplication;
import java.io.IOException;
import javax.servlet.http.*;
@SuppressWarnings("serial")
public class GAEApplicationServlet extends HttpServlet {
    public void doGet(HttpServletRequest req, HttpServletResponse resp)
        throws IOException {
        resp.setContentType("text/plain");
        resp.getWriter().println("Hello, world");
    }
}

```

Сгенерированная страница `index.html` приветствия приложения каталога `war` GAE-проекта содержит ссылку на относительный URL-адрес вызова сервлета, указанный в теге `<url-pattern>` дескриптора `web.xml` (листинг 1.2).

Листинг 1.2. Код страницы `index.html` приветствия GAE-приложения

```
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=UTF-8">
  <title>Hello App Engine</title>
</head>
<body>
  <h1>Hello App Engine!</h1>
  <table>
    <tr>
      <td colspan="2" style="font-weight:bold;">Available Servlets:</td>
    </tr>
    <tr>
      <td><a href="gaeapplication">GAEApplication</a></td>
    </tr>
  </table>
</body>
</html>
```

Папка `src\META-INF` содержит конфигурационный файл `jdbcconfig.xml`, определяющий для приложения конфигурацию механизма Java Data Objects (JDO) сохранения объектов в базе данных.

Каталог `war` GAE-проекта содержит все файлы, необходимые для развертывания приложения в Servlet-контейнере, поэтому в папку `war\WEB-INF\lib` включены библиотеки программного интерфейса API GAE-служб.

Конфигурационный файл `appengine-web.xml` папки `war\WEB-INF` GAE-проекта содержит настройки развертывания приложения на GAE-платформе. Файл `appengine-web.xml` указывает зарегистрированный GAE-идентификатор приложения, версию приложения, а также его список статических и ресурсных файлов и другие настройки работы приложения.

Изначально сгенерированный файл `appengine-web.xml` в теге `<application>` содержит пустой GAE-идентификатор, и для загрузки приложения в облако платформы GAE необходимо заполнить тег `<application>` GAE-идентификатором, полученным в результате регистрации приложения с помощью GAE-консоли **<https://appengine.google.com/>**.

Версия приложения указывается в теге `<version>` файла `appengine-web.xml` и необходима для создания новой версии приложения в облаке GAE-платформы или для замены приложения с тем же номером версии.

Для повышения эффективности GAE-платформа обслуживает отдельно статические файлы приложения, такие как изображения, CSS-стили, JavaScript-файлы

и др., которые могут обслуживаться простым HTTP-сервером, а не сервером приложений, вызывающим сервлеты и обрабатывающим JSP-страницы.

По умолчанию все файлы каталога war приложения, за исключением JSP-файлов и файлов папки WEB-INF, рассматриваются GAE-платформой как статические файлы. Отдельно указать статические файлы можно в файле appengine-web.xml с помощью тега:

```
<static-files>
  <include path="" expiration="" />
  <exclude path="" />
</static-files>
```

Здесь тег `<include>` указывает путь для включения файлов в список статических файлов, а тег `<exclude>` — путь для исключения файлов из списка статических файлов приложения. Атрибут `expiration` указывает время кэширования браузером статических файлов (по умолчанию 10 мин) в формате `d` (дни) `h` (часы) `m` (минуты) `s` (секунды).

Тег `<public-root>` файла `appengine-web.xml` позволяет переопределить корневой каталог для статических файлов приложения.

Все остальные файлы WAR-каталога приложения считаются ресурсными файлами и обслуживаются сервером приложений. Уточнить список ресурсных файлов приложения можно в файле `appengine-web.xml` с помощью тега:

```
<resource-files>
  <include path="" />
  <exclude path="" />
</resource-files>
```

Здесь тег `<include>` указывает путь для включения файлов в список ресурсных файлов, а тег `<exclude>` — путь для исключения файлов из списка ресурсных файлов приложения.

Указать кодировку приложения по умолчанию можно в файле `appengine-web.xml` с помощью тега:

```
<env-variables>
  <env-var name="DEFAULT_ENCODING" value="" />
</env-variables>
```

Файл `appengine-web.xml` может содержать также следующие теги.

◆ `<system-properties>` — посредством дочерних тегов `<property name="" value="" />` определяет системные свойства, которые могут быть доступны из Java-кода с помощью объекта `java.util.Properties`, получаемого методом `System.getProperties()`. Сгенерированный файл `appengine-web.xml` содержит свойство `java.util.logging.config.file` со значением `WEB-INF/logging.properties`, указывающим расположение конфигурационного файла для модуля ведения журнала `java.util.logging`. В конфигурационном файле `logging.properties` установлен уровень ведения журнала по умолчанию — `.level = WARNING`. Просматривать журнал приложения можно в консоли администрирования <https://>

appengine.google.com/ в разделе **Main | Logs**. Выводить свои сообщения в журнал можно с помощью Java-кода:

```
private static final Logger log = Logger.getLogger([имя класса].class.getName());
log.info("");
log.warning("");
log.severe("");
```

Кроме того, GAE-платформа сама устанавливает ряд системных свойств при запуске приложения и JVM:

```
com.google.appengine.runtime.environment,
com.google.appengine.runtime.version, com.google.appengine.application.id,
file.separator, path.separator, line.separator, java.version, java.vendor,
java.vendor.url, java.class.version, java.specification.version,
java.specification.vendor, java.specification.name, java.vm.vendor, java.vm.name,
java.vm.specification.version, java.vm.specification.vendor,
java.vm.specification.name, user.dir.
```

- ◆ `<ssl-enabled>` — с помощью значения `false` отключает использование для приложения протокола HTTPS.
- ◆ `<sessions-enabled>` — с помощью значения `true` включает использование сессий. При этом с помощью тега `<async-session-persistence enabled="true"/>` можно определить асинхронную запись данных сессии в хранилище.
- ◆ `<inbound-services>` — посредством дочерних тегов `<service>[служба]</service>` включает использование GAE-служб:
 - `channel_presence` — приложение получает уведомления о присоединении или отсоединении клиента от Channel-канала;
 - `mail` — приложение может получать электронные письма;
 - `xmpp_message` — приложение может получать мгновенные сообщения. При этом включение службы `xmpp_presence` позволяет приложению получать уведомления об изменении статуса клиента в службе XMPP, включение службы `xmpp_subscribe` — получать уведомления об изменении статуса подписки на сообщения между клиентом и приложением, включение службы `xmpp_error` позволяет приложению получать сообщения об ошибках XMPP-службы;
 - `warmup` — служба посылает запрос для предварительной инициализации нового экземпляра приложения до получения им клиентского запроса, что сокращает время ожидания клиентом ответа от нового экземпляра приложения. GAE-платформа создает новый экземпляр приложения, например, при увеличении количества клиентских запросов к существующему экземпляру приложения.
- ◆ `<warmup-requests-enabled>` — с помощью значения `false` отключает `warmup`-запросы.
- ◆ `<precompilation-enabled>` — с помощью значения `false` отключает оптимизацию загрузки классов. Служба `warmup` работает с приложением, имеющим постоян-

ный трафик с клиентом. Если же клиент посылает запрос к неработавшему приложению, этот запрос, называемый `loading`-запросом, инициализирует экземпляр приложения, и клиент будет ожидать некоторое время ответ на свой первый запрос. `Precompilation`-оптимизация ускоряет работу `loading`-запроса как минимум на 30%.

- ◆ `<admin-console>` — с помощью дочерних тегов `<page name="" url="" />` добавляет страницы администрирования приложением в консоль администрирования <https://appengine.google.com/>.
- ◆ `<static-error-handlers>` — с помощью дочерних тегов `<handler file="" error-code="" />` переопределяет страницы отображения ошибки клиенту по умолчанию, где код ошибки `error-code` может быть `over_quota`, `dos_api_denial` и `timeout`.
- ◆ `<threadsafe>` — с помощью значения `true` включает доставку параллельных запросов к серверу.

Перед запуском GAE-приложения из среды Eclipse удостоверимся, что версия Java раздела **Project Facets** диалогового окна свойств GAE-проекта соответствует версии среды JRE проекта. Для этого в окне **Project Explorer** щелкнем правой кнопкой мыши на узле проекта, в контекстном меню выберем команду **Properties** и раздел **Project Facets** ее диалогового окна (рис. 1.8).

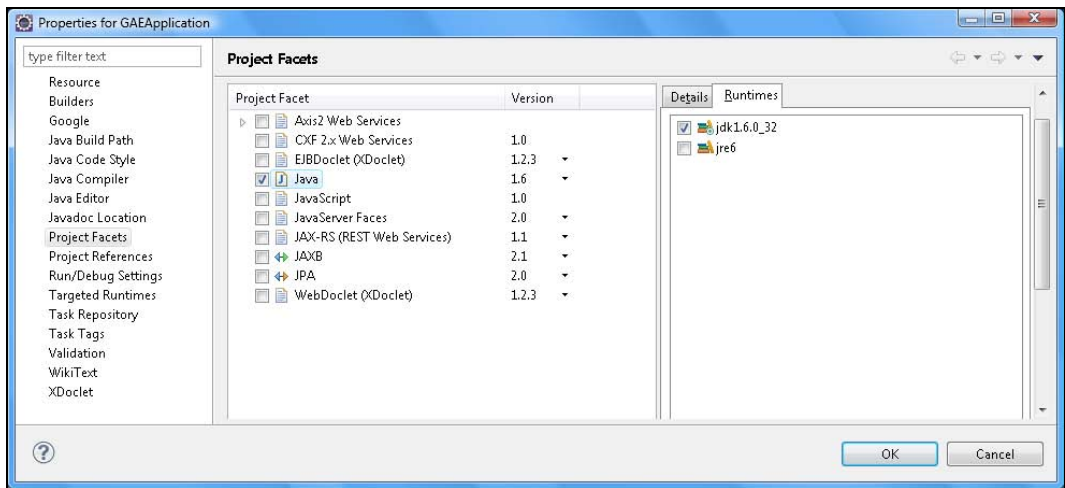


Рис. 1.8. Установка версии Java проекта GAE-приложения

Запуск приложения из среды Eclipse

Для запуска GAE-приложения из среды Eclipse в окне **Project Explorer** щелкнем правой кнопкой мыши на узле проекта и в контекстном меню выберем команду **Run As | Web Application**. В результате в окно **Console** среды Eclipse будет выведено сообщение:

```

... AM com.google.apphosting.utils.jetty.JettyLogger info
INFO: Logging to JettyLogger(null) via com.google.apphosting.utils.jetty.JettyLogger
... AM com.google.apphosting.utils.config.AppEngineWebXmlReader readAppEngineWebXml
INFO: Successfully processed ... \workspace\GAEApplication\war\WEB-INF/appengine-web.xml
... AM com.google.apphosting.utils.config.AbstractConfigXmlReader readConfigXml
INFO: Successfully processed ... \workspace\GAEApplication\war\WEB-INF/web.xml
... PM com.google.appengine.tools.development.DevAppServerImpl start
INFO: The server is running at http://localhost:8888/
... com.google.appengine.tools.development.DevAppServerImpl start
INFO: The admin console is running at http://localhost:8888/_ah/admin

```

В выведенном сообщении указан адрес локального вызова приложения **http://localhost:8888/** и адрес локальной консоли администрирования **http://localhost:8888/_ah/admin**.

Набрав в адресной строке Web-браузера адрес **http://localhost:8888/**, можно будет увидеть страницу приветствия приложения со ссылкой вызова сервлета приложения (рис. 1.9).

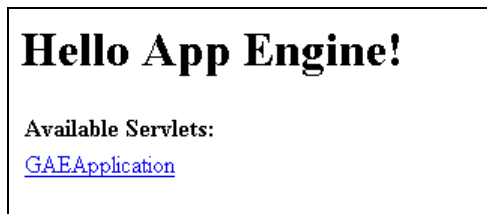


Рис. 1.9. Страница приветствия GAE-приложения

Набрав в адресной строке Web-браузера адрес **http://localhost:8888/_ah/admin**, можно будет увидеть страницу консоли администрирования (рис. 1.10).

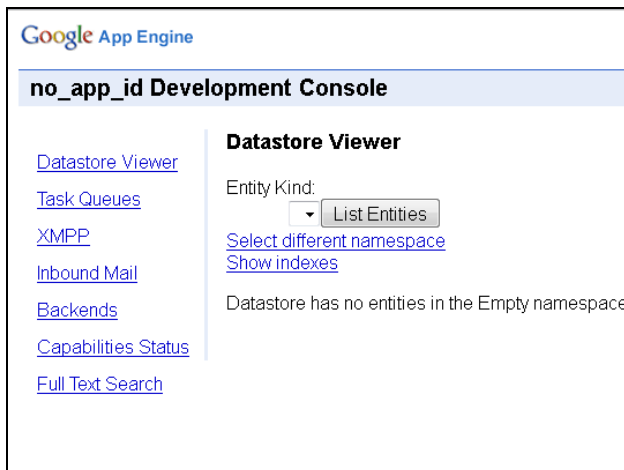


Рис. 1.10. Страница локальной консоли администрирования GAE-приложения

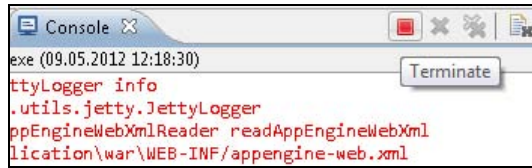


Рис. 1.11. Остановка GAE-приложения кнопкой **Terminate**

Остановить работающий сервер с GAE-приложением можно, нажав кнопку **Terminate** окна **Console** среды Eclipse (рис. 1.11).

Развертывание приложения на платформе App Engine

Регистрация приложения

Для того чтобы развернуть приложение на платформе GAE, его необходимо предварительно зарегистрировать. Для регистрации GAE-приложения откроем Web-браузер и попробуем войти в консоль администрирования по адресу **https://appengine.google.com/**. Однако сервис консоли администрирования является защищенным сервисом и его аутентификацию/авторизацию проходят лишь пользователи, имеющие свой Google-аккаунт. Поэтому предварительно откроется страница, приглашающая ввести логин/пароль Google-аккаунта или зарегистрировать его (рис. 1.12).

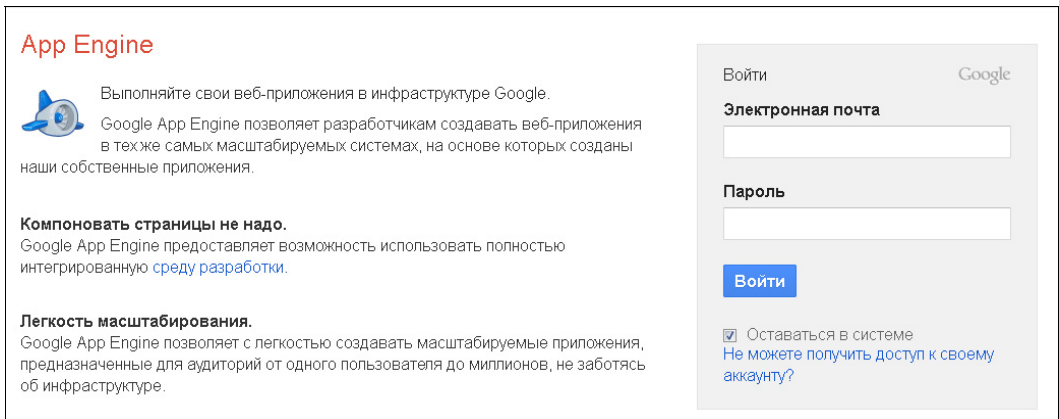


Рис. 1.12. Страница аутентификации консоли администрирования

После ввода логина/пароля и нажатия кнопки **Войти** откроется страница консоли администрирования.

Кнопка **Create Application** консоли администрирования позволяет зарегистрировать новое приложение. Страница, открывающаяся при нажатии кнопки **Create Application**, отображает уведомление об оставшемся количестве приложений, ко-

торые можно еще зарегистрировать на данный Google-аккаунт: максимальное число — 10. Поле **Application Identifier** предлагает ввести идентификатор приложения, при этом кнопка **Check Availability** позволяет проверить его уникальность. Поле **Application Title** предлагает ввести отображаемый при аутентификации заголовок приложения.

Раздел **Authentication Options (Advanced)** страницы регистрации GAE-приложения с помощью переключателей позволяет выбрать систему аутентификации пользователей, которая может быть использована GAE-приложением:

- ◆ **Open to all Google Accounts users (default)** — аутентификация с помощью Google-аккаунта;
- ◆ **Restricted to the following Google Apps domain** — аутентификация с помощью аккаунта домена бизнес-приложения Google Apps;
- ◆ **(Experimental) Open to all users with an OpenID Provider** — аутентификация с помощью идентификатора OpenID.

Раздел **Storage Options (Advanced)** страницы регистрации GAE-приложения с помощью переключателей **High Replication (default)** и **Master/Slave** позволяет выбрать тип хранилища данных приложения — распределенное хранилище или главное/подчиненное хранилище (в настоящее время не рекомендуется). Данный раздел после отмены хранилища Master/Slave был исключен.

Нажатие кнопки **Create Application** страницы регистрации GAE-приложения завершает регистрацию приложения.

Теперь при входе в консоль администрирования <https://appengine.google.com/> ее страница будет содержать ссылку на зарегистрированное приложение, при щелчке по которой откроется страница администрирования приложением.

Страница администрирования приложением

Левая панель страницы администрирования приложением содержит разделы **Main**, **Data**, **Administration**, **Billing** и **Resources**.

Раздел **Main** содержит подразделы **Dashboard**, **Instances**, **Logs**, **Versions**, **Backends**, **Cron Jobs**, **Task Queues**, **Quota Details**.

Раздел **Dashboard** показывает общую информацию о работе приложения в App Engine, включающую в себя набор графиков потребления ресурсов приложением, информацию о работающих экземплярах приложения, статус оплаты хостинга приложения, таблицу потребления ресурсов приложением в пределах установленной квоты (<https://developers.google.com/appengine/docs/quotas>), информацию о URI-запросах к приложению и об их ошибках. Набор графиков потребления ресурсов приложением состоит из следующих графиков:

- ◆ **Requests/Second** — количество запросов в секунду в зависимости от времени, включая динамические, статические и кэшированные запросы;
- ◆ **Requests by Type/Second** — количество запросов в секунду в зависимости от времени с дифференциацией по типам запросов: динамическим, статическим (к статическим ресурсам приложения) и кэшированным;