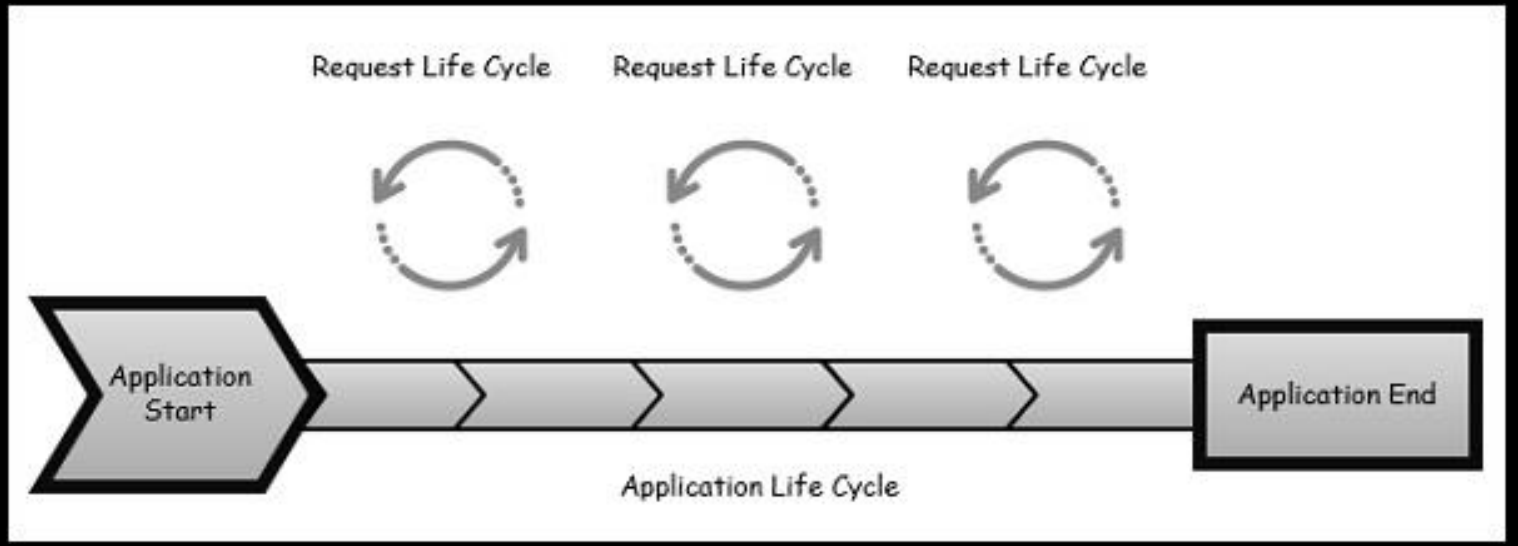
ASP. Net MVC

Back 2 school

TRADING TOOLS

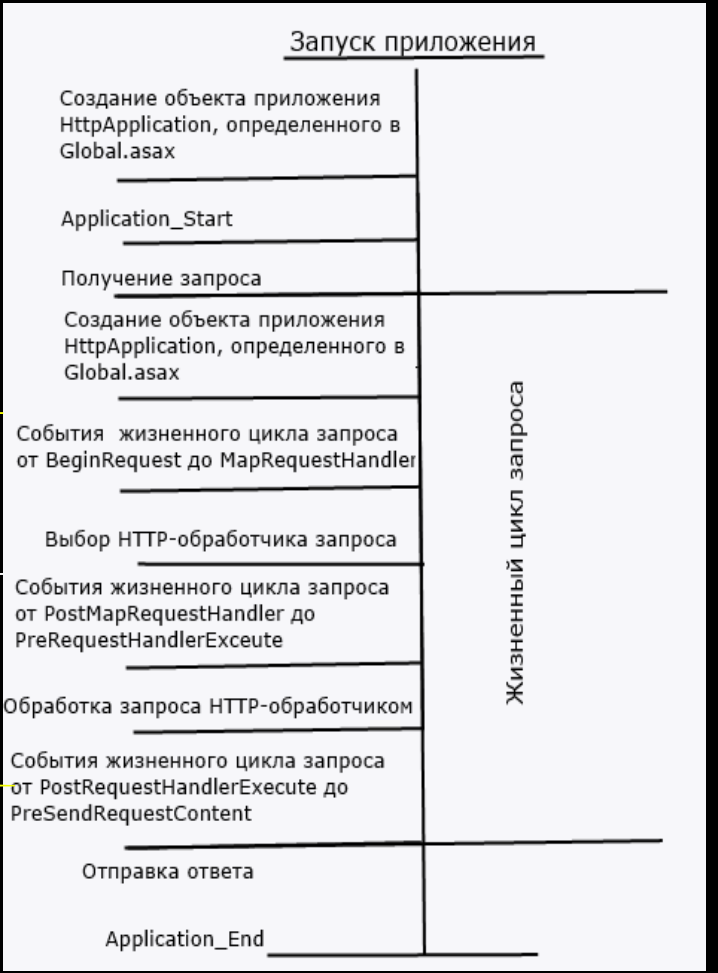
APPLICATION LIFE CYCLE



APPLICATION LIFE CYCLE

1. BeginRequest: когда приложение получает новый запрос
2. AuthenticateRequest/PostAuthenticateRequest: идентификация (аутентификация) пользователя
3. AuthorizeRequest/PostAuthorizeRequest: авторизация
4. ResolveRequestCache/PostResolveRequestCache: при получении данных из кэша
5. MapRequestHandler/PostMapRequestHandler: определении обработчика запроса
6. AquireRequestState/PostAquireRequestState: при получении данных состояния, связанных с запросом (например, данные сессии)
7. PreRequestHandlerExecute/PostRequestHandlerExecute: перед/после работы обработчика запроса
8. ReleaseRequestState/PostReleaseRequestState: перед/после освобождения данных, ассоциированные с запросом
9. UpdateRequestCache: обновлении данных в кэше
10. LogRequest/PostLogRequest: перед/после каждым логгированием
11. EndRequest: когда данные для ответа уже готовы к отправке клиенту
12. PreSendRequestHeaders: перед отправкой HTTP-заголовков браузеру клиента
13. PreSendRequestContent: после отправки заголовков, но перед отправкой основного содержимого ответа

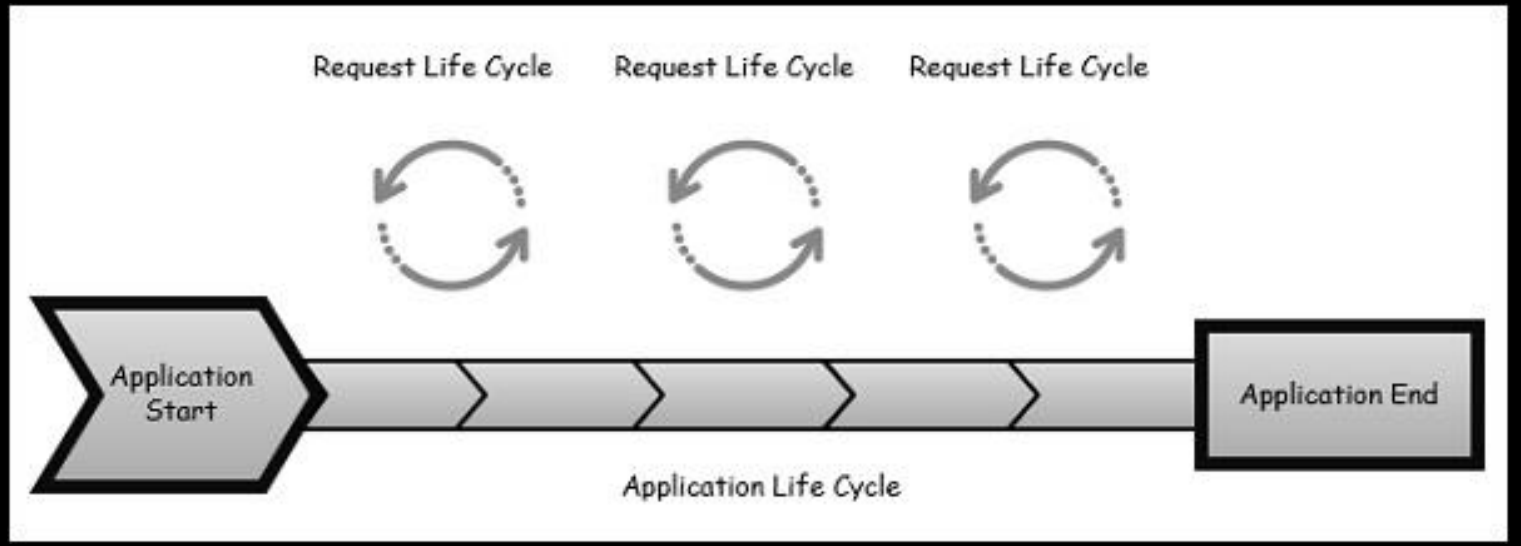
Error



DEMO

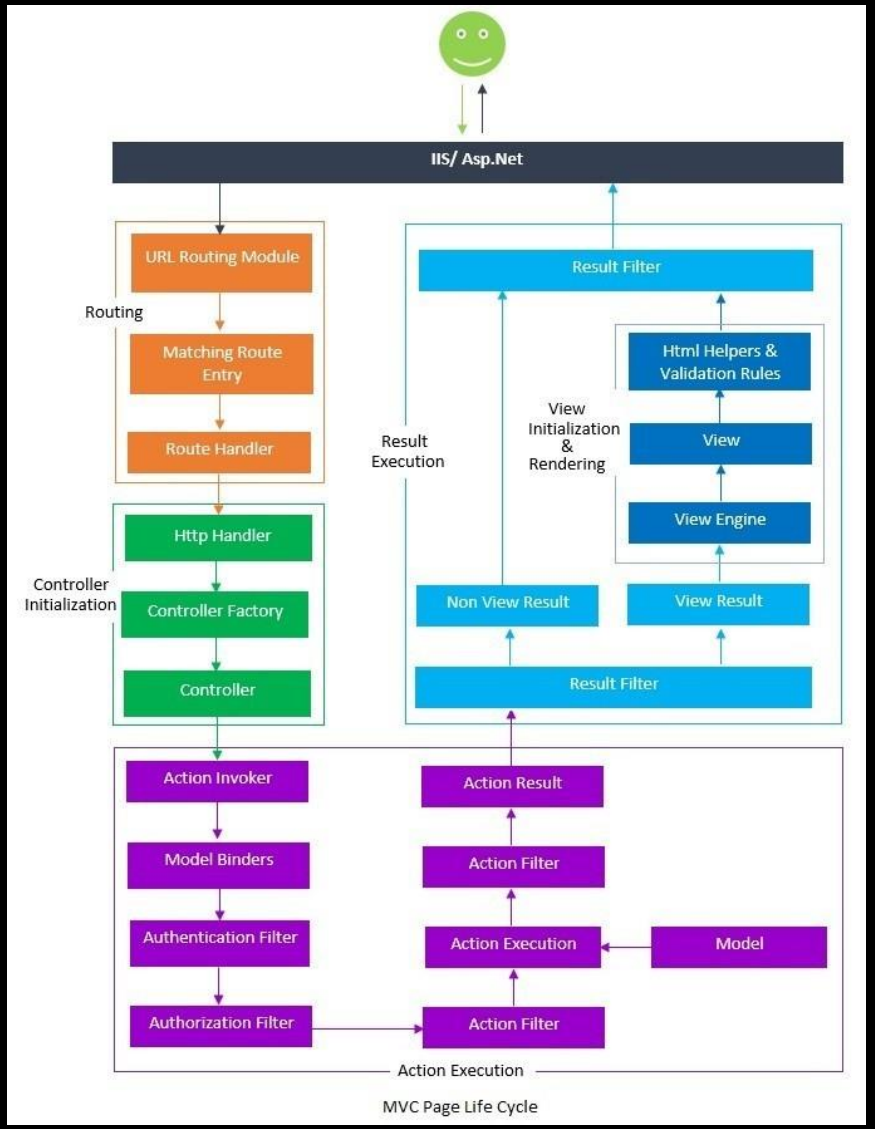
TRADING TOOLS

Application Life Cycle



What is the request life cycle?

Request Life cycle



MVC Page Life Cycle

HTTP Module

HTTP модули выполняются до и после обработки и обеспечивают метод для взаимодействия с запросом. Пользовательские модули должны быть унаследованы от System.Web.IHttpModule интерфейса.

```
void Init(HttpApplication context);
void Dispose();
```

Модули, как правило, синхронизированы с событиями System.Web.IHttpModule класса в рамках Global.asax.cs

Зарегать модуль нужно к web.config

```
<system.webServer>
 <modules>
  <add name="Timer" type="LifeCycleApp.Modules.TimerModule"/>
 </modules>
</system.webServer>
```

1. BeginRequest
2. AuthenticateRequest
3. AuthorizeRequest
4. ResolveRequestCache
5. AcquireRequestState
6. PreRequestHandlerExecute
7. PostRequestHandlerExecute
8. ReleaseRequestState
9. UpdateRequestCache
10. EndRequest
11. PreSendRequestHeaders *
12. PreSendRequestContent *

Error *

* могут произойти в любое время в течение запроса, все остальные перечислены в порядке их вызова.

DEMO

TRADING TOOLS

HTTP Handler

Используются для генерации ответа на HTTP-запрос. При обработке одного запроса мы можем задействовать несколько различных модулей, но только один HTTP Handler. Пользовательские хендлеры должны быть унаследованы от System.Web.IHttpHandler интерфейса.

```
void ProcessRequest(HttpContext context);
bool IsReusable { get; }
```

HTTP Handler, как правило, регистрируются при формировании RouteConfig в качестве обработчика к маршруту. Провайдером для хендлера является IRoteHandler

Зарегать хендлер нужно к web.config

```
<system.webServer>
 <handlers>
  <add name="MyHttpHandler" path="/handler/" verb="GET"
type="LifeCycleApp.Handlers.MyHttpHandler"/>
 </handlers>
</system.webServer>
```

1. BeginRequest
2. AuthenticateRequest
3. AuthorizeRequest
4. ResolveRequestCache

--- MapRequestHandler (selecting handler) ---

5. AcquireRequestState
6. PreRequestHandlerExecute

--- executes HTTP Handler ---

5. PostRequestHandlerExecute
6. ReleaseRequestState
7. UpdateRequestCache
8. EndRequest
9. PreSendRequestHeaders *
10. PreSendRequestContent *

Error *

Мы также можем поставить любой тип/глагол/verb запроса - в этом случае используется звездочка *

DEMO

TRADING TOOLS

Handler vs Module

Modules

1 запрос может быть обработан несколькими модулями
Создан для изменения или передачи запроса остальным модулям
Реализует IHttpModule интерфейс
Создан что бы быть интегрированными с любимы событием из жизненного цикла

Handlers

Только 1 хендлер может обработать запрос
Отвечает за формирует ответ и возвращает
Реализует IHttpHandler интерфейс
Связан с событиями выбора хенделра и обработки запроса

Регистрируется в коде или в web.config файле

FILTERS

Тип фильтров	Реализуемый интерфейс	Стандартная реализация	Описание
Фильтры аутентификации	IAuthenticationFilter	-	Фильтр, определяющий, аутентифицирован ли клиент. Данный фильтр запускается до выполнения любого другого фильтра или метода действий
Фильтры авторизации	IAuthorizationFilter	AuthorizeAttribute	Фильтр, определяющий, имеет ли пользователь доступ к данному ресурсу. Данный фильтр запускается после фильтра аутентификации, но до любого другого фильтра или метода действия
Фильтры действий	IActionFilter	ActionFilterAttribute	Фильтр, применяемый к действиям. Может запускаться как до, так и после выполнения метода действий
Фильтры результатов действий	IResultFilter	ActionFilterAttribute	Фильтр, применяемый к результатам действий. Может запускаться как до, так и после выполнения результата действия
Фильтры исключений	IExceptionFilter	HandleErrorAttribute	Атрибут для обработки исключений, выбрасываемых методом действий и результатом действий

DEMO

TRADING TOOLS

Q&A

TRADING TOOLS