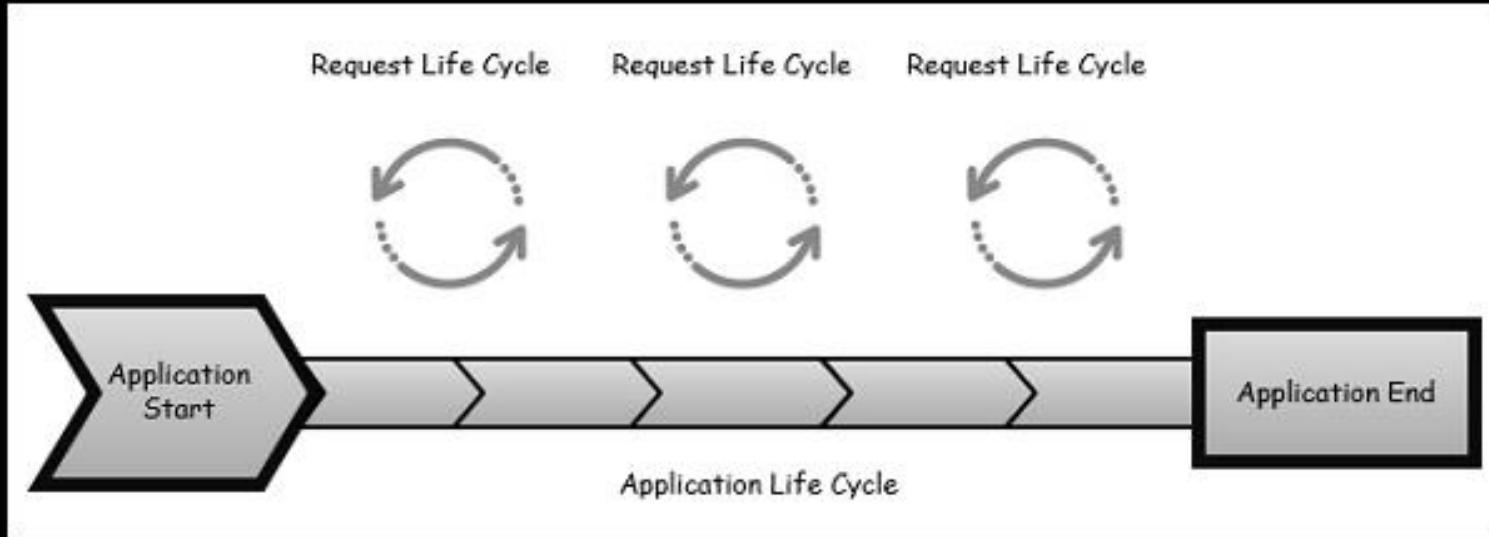


ASP. Net MVC

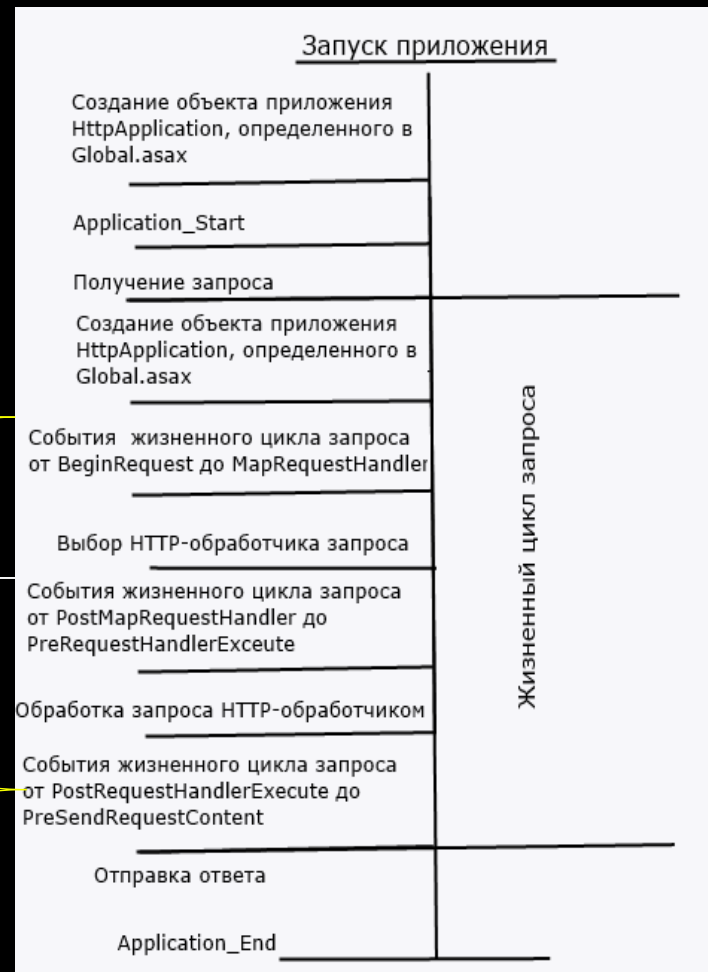
Back 2 school

APPLICATION LIFE CYCLE



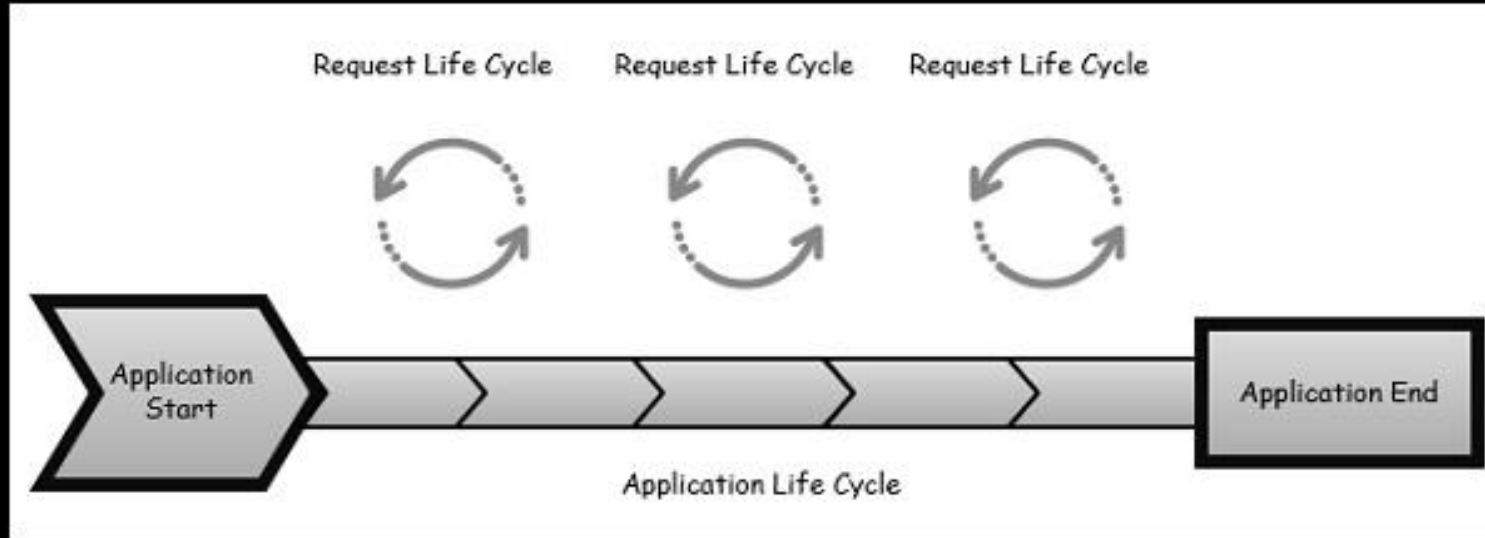
APPLICATION LIFE CYCLE

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 логированием
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



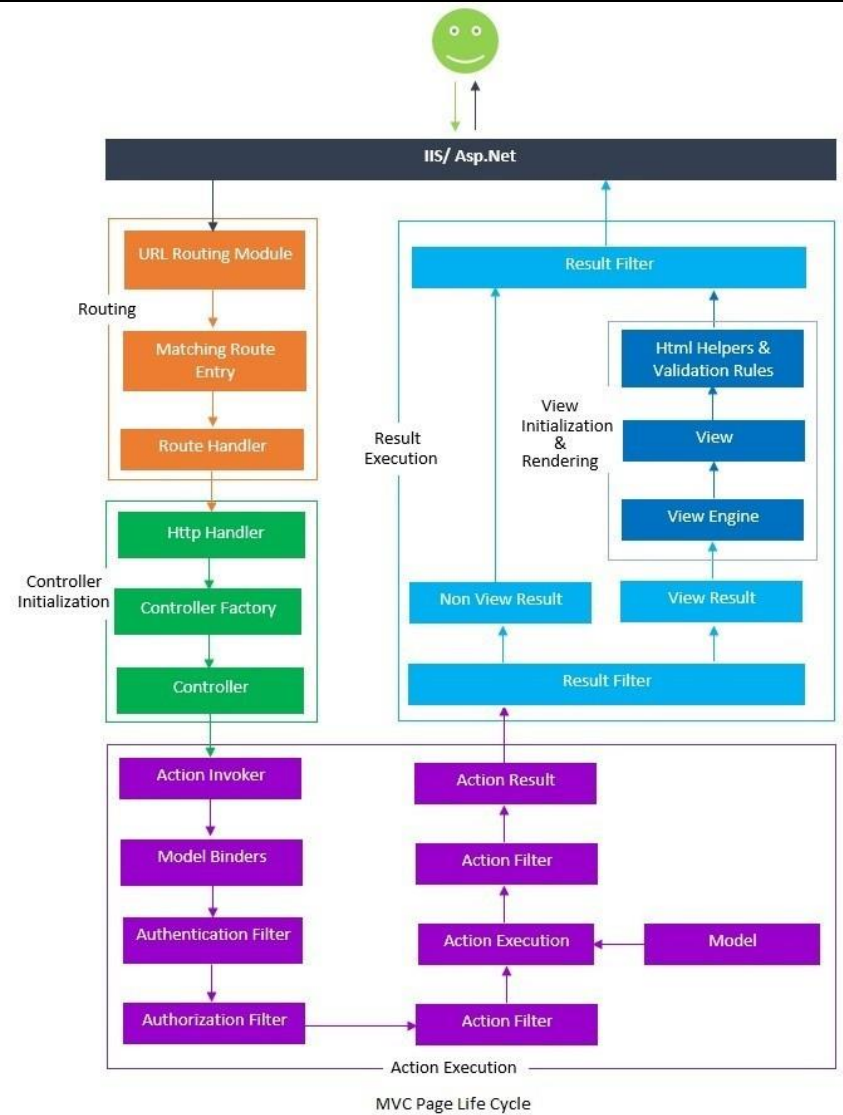
DEMO

Application Life Cycle



What is the request life cycle?

Request Life cycle



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**

Зарегатать 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 *
13. Error *

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

DEMO

