После получения IIS запроса на обработку, при первом обращении к ресурсу, работающему под управлением CLR, создается объект класса **ApplicationManager**, который представляет домен приложения, внутри которого обрабатывается запрос. Домены приложений изолируют выполняющиеся приложения друг от друга. Уже внутри домена приложения создается объект класса **HostingEnvironment**, который предоставляет доступ к информации о приложении, в частности, он сообщает имя и каталог приложения.

объекты, как **HttpContext, HttpRequest** и **HttpResponse**, которые инкапсулируют всю информацию, связанную с текущим запросом к приложению.

экземпляр класса **HttpApplication**. Если в приложении определен файл Global.asax, то среда ASP.NET в качестве приложения создает объект класса Global.asax, который в свою очередь наследуется от класса **HttpApplication**. На этой же стадии происходит начальная инициализация приложения в методе **Application**\_**Start**,

После этого запрос начинает обрабатываться в конвейере класса **HttpApplication**. Запрос перехватывается специальным HTTP-модулем под названием **UrlRoutingModule**. Этот модуль выбирает маршрут, который соответствует входящему запросу.

А сама регистрация маршрутов происходит в методе **Application\_Start** строкой RouteConfig.RegisterRoutes(RouteTable.Routes);. Для каждого приложения имеется только один объект коллекции маршрутов RouteTable.

объект **IRouteHandler**. По умолчанию объект IRouteHandler представляет экземпляр класса **MvcRouteHandler**.

объекта IRouteHandler вызывается метод **GetHandler**, который возвращает объект интерфейса **IHttpHandler**. По умолчанию в качестве IHttpHandlera используется объект класса **MvcHandler**.

У обработчика **IHttpHandler** вызывается метод **ProcessRequest** для обработки запросов. создание контроллера, который представляет объект интерфейса **IController**. За создание контроллера отвечает объект интерфейса **IControllerFactory**, представляющий фабрику контроллеров.

Vыполнение кода контроллера. После инициализации для запуска метода для обработки запроса контроллер вызывает метод **InvokeAction**():

Далее привязчик модели (по умолчанию класс System.Web.Mvc.DefaultModelBinder) извлекает данные из запроса, производит из преобразование, форматирование, валидацию и связывает их с определенными параметрами вызываемого метода.

**IAuthenticationFilter**

**IAuthorizationFilter**

**OnActionExecuting** фильтра действий. Фильтр действий представляет объект интерфейса **IActionFilter**. Кроме того, также после выполнения метода контроллера запускается другой метод фильтра действий - метод **OnActionExecuted**

результат обработки в виде объекта **ActionResult**.

объект интерфейса **IResultFilter**. Его метод **OnResultExecuting** срабатывает до обработки результата, а метод **OnResultExecuted** после.

результатов действий - **ViewResult, ContentResult, FileResult, RedirectResult** и др.

у каждого объекта **ActionResult** вызывается метод **ExecuteResult**

