Controller 1: Captcha, mit Session

**using** System;

**using** System.Collections.Generic;

**using** System.Linq;

**using** System.Web;

**using** System.Web.Mvc;

**using** SomeProject.CoreSystem.Business.Session;

**using** XCaptcha;

**namespace** SomeProject.CoreSystem.Web.Controllers

{

**public** **class** CaptchaController : Controller

    {

**private** **readonly** ISessionStateAccessor \_sessionStateAccessor;

**public** CaptchaController(ISessionStateAccessor sessionStateAccessor)

        {

            \_sessionStateAccessor = sessionStateAccessor;

        }

        [AcceptVerbs(HttpVerbs.Get)]

**public** ActionResult Captcha(**string** captchaSessionKey, **string** currentCaptcha)

        {

**string** sessionCaptcha;

            ImageBuilder captcha;

**if** (\_sessionStateAccessor.TryGetValue(captchaSessionKey, **out** Captcha) && sessionCaptcha == currentCaptcha)

            {

                captcha = **new** ImageBuilder(sessionCaptcha);

            }

**else**

            {

                captcha = **new** ImageBuilder();

            }

            var result = captcha.Create();

            \_sessionStateAccessor.SetValue(captchaSessionKey, result.Solution);

**return** **new** FileContentResult(result.Image, result.ContentType);

        }

    }

}

Interessante Elemente:

* Session State wird als Abhängigkeit von aussen zur Verfügung gestellt. Kann daher faktisch irgendein Store sein.
* Controller enthält keinerlei fachliche Logik, sondern nur Ablauf