Track: Elevate

ASP.NET MVC API
App: iServe

Tim Hardy and Tudor Lupan
Fellowship Technologies

Vision

I don't know what your destiny will be, but one thing I do know: the only ones among you who will be really happy are those who have sought and found how to serve.

- Albert Schweitzer

dynamic 09
CHURCH

Only a life lived for others is a life worthwhile.

- Albert Einstein

Nobody made a greater mistake than he who did nothing because he could do only a little.

- Edmund Burke

You, my brothers, were called to be free. But do not use your freedom to indulge the sinful nature; rather, serve one another in love

- Galations 5:13 NIV

I can do this

IWantToServe.com

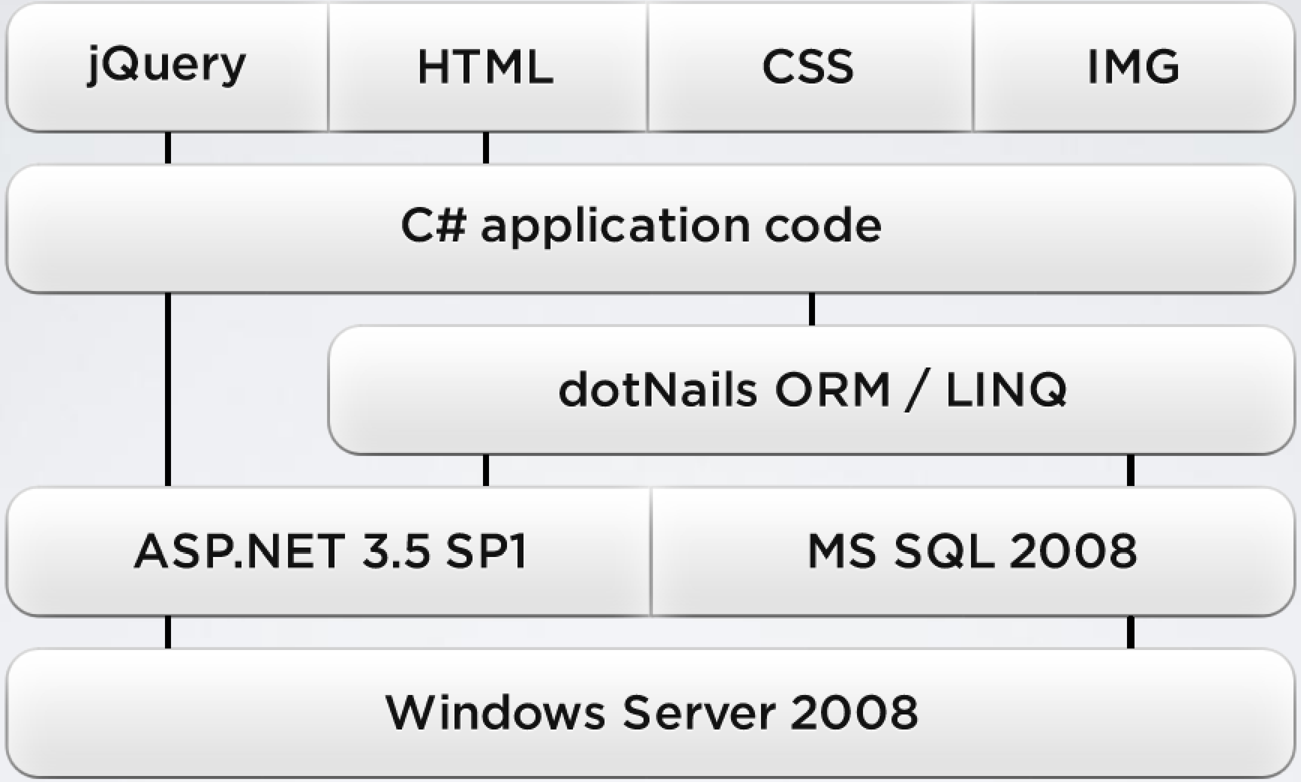
Technologies Used

- ASP.NET MVC 1.0 with MVC Futures
- Visual Studio 2008 with ASP.NET 3.5 SP1
- C# 3.0
- MS SQL Server 2005
- LINQ
- DataAnnotations
- dotNails

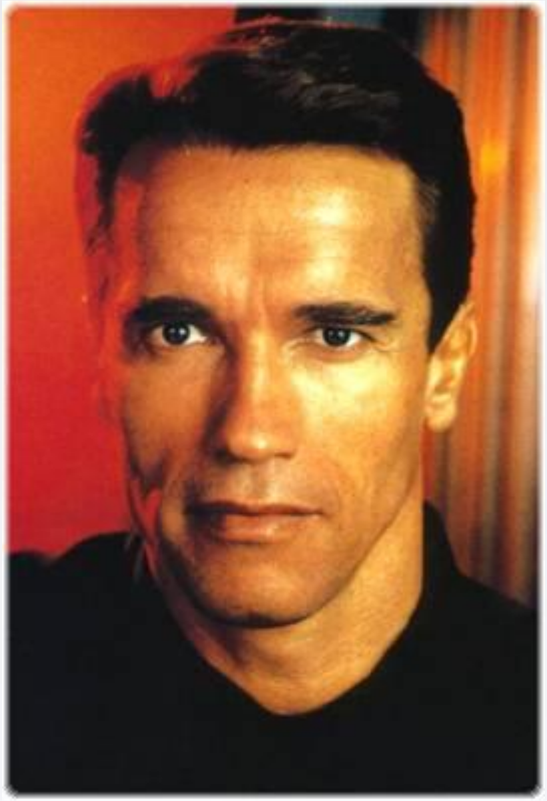
F1 API

- Leverages WebLink logins
  - Our UserID **IS** an F1 PeopleID
- User profile editor
- 3rd Party Consumer
- Custom .NET OAuth client library
- Xml content types

Fellowship Tech Architecture Stack



OAuth



Get a Request Token

```
public ActionResult LogOn() {
     try {
          // Ask the API for a request token, so we can begin talking to the API
          Token requestToken = OAuthUtil.GetRequestToken();

          // Store in Session
          this.RequestToken = requestToken;
          …
```

Authenticate the request token

```
if (requestToken != null) {
             string callbackUrl =
                      string.Format("http://{0}/Account/RequestAccessToken",
                                   Request.Url.Host);

             // Get a link for FT user login page and provide a callback url
             string authorizationLink =
                      OAuthUtil.RequestUserAuth(requestToken.Value,
                                                callbackUrl);

             // Ask the API to log the user in
             return new RedirectResult(authorizationLink);
}
```

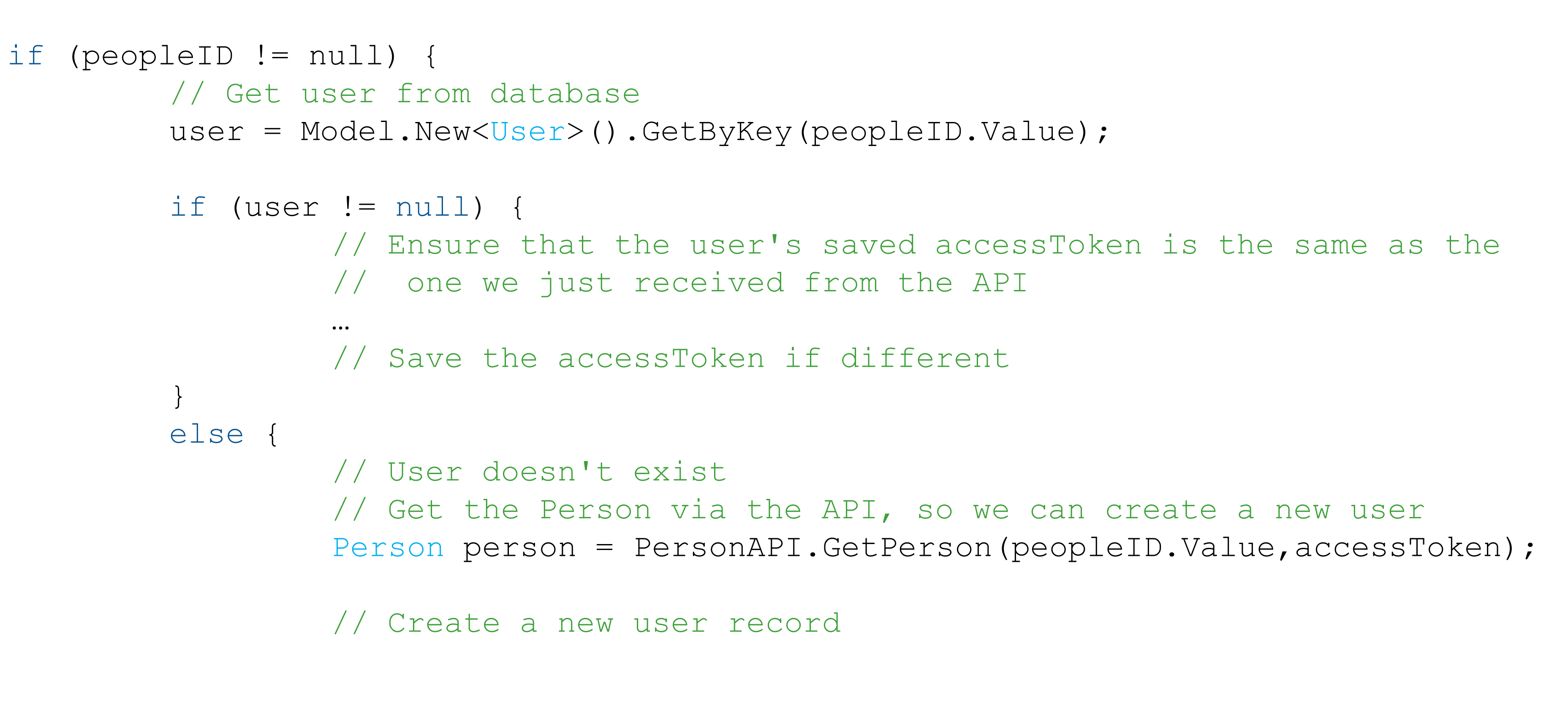
Request Access Token

```
public ActionResult RequestAccessToken() {

         accessToken = OAuthUtil.GetAccessToken(this.RequestToken,
                                               out personUrl);

         string id = personUrl.Substring(personUrl.LastIndexOf("/")+1);
                                        peopleID = Convert.ToInt32(id);
```

Sync User



```
if (peopleID != null) {
        // Get user from database
        user = Model.New<User>().GetByKey(peopleID.Value);

        if (user != null) {
                // Ensure that the user's saved accessToken is the same as the
                //  one we just received from the API
                …
                // Save the accessToken if different
        }
        else {
                // User doesn't exist
                // Get the Person via the API, so we can create a new user
                Person person = PersonAPI.GetPerson(peopleID.Value,accessToken);

                // Create a new user record
```

Access protected resources

```
public Person GetPerson(int personID, Token accessToken) {
        string url =
        string.Format("https://ftapiwater.staging.fellowshiponeapi.com/v1/People/
                    {0}", personID);

        // Create a request to the API to obtain the person.
        HttpWebRequest webRequest = OAuthUtil.CreateWebRequest(url, accessToken,
                                                  HttpRequestMethod.GET);

        using (WebResponse webResponse = webRequest.GetResponse()) {
                using (StreamReader streamReader = new
                      StreamReader(webResponse.GetResponseStream())) {
                      XmlSerializer xmlSerializer = new
                      XmlSerializer(typeof(Person));

                      // Deserialize the response into a Person object.
                      Person person = xmlSerializer.Deserialize(streamReader) as
                                                           Person;
```

LINQ to XML

```
private void PopulateCommunicationValues(Person person, Token accessToken) {
         // Create a request to the API to obtain the communication values.
         WebRequest webRequest = OAuthUtil.CreateWebRequest(
                                  string.Format("{0}/People/{1}/Communications",
                                  baseUrl, person.ID),
                                  accessToken, HttpRequestMethod.GET);

         using (WebResponse webResponse = webRequest.GetResponse()) {
                 using (StreamReader streamReader =
                          new StreamReader(webResponse.GetResponseStream())) {

                 // Load the xml response from the API into an XElement so that
                 //  the communication values can be traversed.
                 XElement xml = XElement.Load(streamReader);

                 if (xml.Element("communication") != null) {
                           var individualEmail = xml.Elements("communication")
                                  .FirstOrDefault(comm =>
                                  !string.IsNullOrEmpty(comm.Element("person")
                                  .Attribute("id").Value) &&
                                  communication.Element("communicationType")
                                  .Value.Equals("Email"));

                 }
         }
```

What do you want to see?

Gotcha’s and speed bumps

- .NET OAuth library is not good
- POST or PUT via Web Request
    - Must use ASCII
    - Can’t use chunking

Security is coming

One act of beneficence, one act of real usefulness, is worth all the abstract sentiment in the world.

- Ann Radcliffe

Please Fill Out The Session Evaluation!

Tim Hardy
thardy@fellowshiptech.com

Tudor Lupan
tlupan@fellowshiptech.com

Questions?

How wonderful it is that nobody need wait a single moment before starting to improve the world

- Anne Frank