

Leveraging Conventions in .NET

Conventions are polices that are encoded in your application/framework/team that guide you toward doing the "right" thing.

They do things automatically (magic!), but can be overridden when necessary

Dates

Show pages of application

- Edit EditPatient.aspx, add Id() to date tags
- write script to attach datepickers to those IDs

Show it works

Explicit coding of Ids is tiresome

- eliminate duplication, create convention to generate IDs, remove Id()s

Refresh, view source, show IDs are updated, datepickers still work

jQuery allows using CSS selectors, clean up the script with a single call

- manually AddClass("date"), update script

Show it still works

We are always going to want datepickers for dates - make it happen!

- create convention to add "date" class based on data type, remove AddClass() calls

Show it still works

Show appointment page, notice dates arent pickers

- move script to Edit.master page, now show both pages have datepickers

Two of the inputs should really be times

- manually AddClass("time"), update script to showTime=true

Show it works (show other date field is still only date)

We will always want times to use timepickers

- create convention to add "datetime" based on property name

Help

git checkout helpstart

Show script needed to call to show context sensitive help

show currently hardcoded link to topic in main navigation

- create html convention to add help link after every input
- write script in site.master that wires up all helpLinks to showHelp using the topic from metadata

Show the app. click on various questions marks. mention that the topic is just a key - the related document will have proper information

- Add a script text/html tag on site.master that contains a HelpLinkTag for the page's topic
- append the link's html() to the first h2

Database

We can use conventions to generate our database

Show empty database

- run irake

Refresh database

Point out table and column names and datatypes generated by convention

- Add State property
- edit EditPatient to add input for State
- run irake

The state column is also 255 - thats too big

Refresh the app, show editing the State allows a lot of characters.

Click Save - succeeds! even without name, and with long state

The model doesn't contain enough information. There are additional details we would like to express - attributes!

- Write attributes for MaxLength and Required
- apply attributes to model
- write FNH convention to respect attributes
- run irake

Refresh the database, show the columns changed

Show app is unchanged

- Write html convention for maxlength and required
- build

Show app - State will not allow more than 2 characters

Save - should get validation errors about name

- bonus! Dates arent validated add convention for "date" class