STONYPOINT

Advanced Force.com Development (SPDEV-503)

Author: Steve Wasula / Jeremy White

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Agenda

STONYPOINT

- **Day 1**
  - Agenda & Course Overview
  - Introduction to GenWatt
  - Setting up a Practice Site
  - Setting up the Force.com IDE
  - Salesforce Security Review
  - Working with the Force.com IDE
  - SOQL
  - Model – View – Controller
- **Day 2**
  - Declarative Programming Review
  - Apex Fundamentals
    - Syntax
    - Structure
    - Patterns
    - DML
  - Developing Apex Triggers
    - Updating Records
    - Creating Records
    - Sharing Records
  - Debugging

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Agenda

STONYPOINT

- **Day 3**
  - Apex Classes
    - Syntax
    - Unit Testing
  - Visualforce fundamentals
    - Syntax
    - Structure
  - Developing Visualforce Pages & Controllers
    - Controllers (Standard, Extensions, Custom, Standard Set)
    - Custom Controllers
    - Controller Extensions
  - Inserting Visualforce into the User Interface
  - Sending emails using Apex and Workflow
  - Using transition pages
  - Batch & Scheduled Apex
  - Deployment & Change Management
  - Session Wrap-up

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Agenda

STONYPOINT

- **Day 4**
  - Advanced Visualforce
    - Dynamic, dependent picklists
    - Visualforce Components
      - Developing
      - Passing & consuming parameters
    - Visualforce in Dashboards
    - Charting
    - Building a Content Management System with Visualforce
  - Integrations Overview
  - Salesforce Integration Technologies
    - REST API's
    - SOAP API's
    - Salesforce to Salesforce
  - Security, Authorization and Authentication
    - Force.com Security
    - Remote Sites
    - Single Sign-on
    - OAuth2
  - Developer Workbench

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Agenda

STONYPOINT

- **Day 5**
  - Force.com SOAP API
    - Overview of the SOAP API
    - Retrieving Data with SOQL
    - Enterprise WSDL
    - Partner WSDL
    - Developing a custom SOAP Web Service
    - Consuming a custom SOAP Web Service
  - REST Overview
  - JSON Basics
  - Force.com REST API
    - Overview of the REST API
    - Retrieving Data with SOQL
    - Developing a custom REST API
    - Consuming a custom REST API
      - Parsing JSON
  - Force.com Streaming API
    - Overview of the Streaming API
    - Consuming the Streaming API
  - Email Services
    - Developing an Email Service
    - Consuming an Email Service
  - Force.com Sites
    - Developing a Force.com Site
    - Developing an unauthenticated REST Web Service using Visualforce
    - Consuming an unauthenticated REST Web Service

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Course Overview

STONYPOINT

- About this Course
- About the Instructor
- Introductions
- Course Structure
  - Introduce concept & technology
  - Exercise to use concept in system
- Housekeeping

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Business Scenario: GenWatt

STONYPOINT

GenWatt sells Generators to both businesses and consumers and uses Salesforce for sales force automation, customer services and marketing activities. GenWatt wants to utilize the full power of the platform to automate as many processes as possible. GenWatt's technology leadership wants to use declarative technology whenever possible and only utilize programmatic solutions when the business need cannot be addressed using declarative technology.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Your Practice Site

STONYPOINT

For this hands-on training class, you will need to set up a **Salesforce Developer Edition** to perform the class exercises.

The **Salesforce Developer Edition** is a completely free testing environment that contains a minimal amount of test data and will be available to you forever.

Perform the following steps to set up a Developer Edition for hands-on exercises in this class.

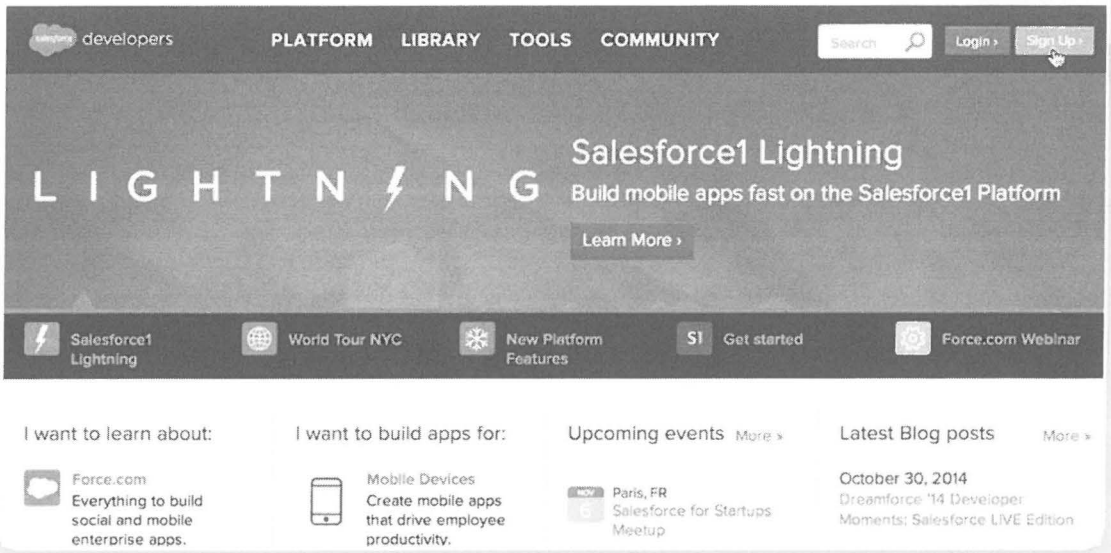
Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Practice Site Action Steps

STONYPOINT

1. Go to http://developer.salesforce.com
2. Click on the green "Sign Up" button



Copyright © 2012 - 2014 Stony Point, Inc. All rights reserved.

Copyright © 2012 - 2014 Stony Point, Inc. All rights reserved.

Practice Site Action Steps

STONYPOINT

3. Fill out the form provided to set up your developer's edition and click ***Submit***

*Username Format: **firstname.lastname@spdev205.dev**

Copyright © 2012 - 2014 Stony Point, Inc. All rights reserved.

Copyright © 2012 - 2014 Stony Point, Inc. All rights reserved.

Practice Site Action Steps

STONYPOINT

4. Check your email inbox associated with your developer edition
5. Click on the link within the email from Salesforce to login to your developer edition

Welcome to Force.com Developer Edition.
Dear Dana Cowden,

Your user name is below. Note that it is in the form of an email address:

User name: dana@training.com.dev

You'll be asked to set a password and password question and answer when you first log in.
Passwords are case sensitive.
Your password question and answer will be used if you forget your password. Make sure to choose a password question and answer that you will easily remember.

Click https://login.salesforce.com?c=l14m45n2Bps;q4eFXCNzL7Ej5R3swJzRTgaSIfMZadIqiptjd6.jSLjh13hOwgvI4ISJRmmKklmq%3D4Sk to log in now.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Practice Site Action Steps

STONYPOINT

6. Create your new password and security question
7. Click ***Save*** to log in to your new developer edition

Welcome to salesforce.com!

For security reasons, your password must now be changed. Your password was last changed or reset on 8/13/2013 7:57 AM

User Name admin@sptl101.dev

New Password ••••••••••

Strong Your password is strong. It contains at least 8 alphanumeric characters, and includes uppercase and lowercase.

Verify New Password ••••••••••

Question What is your pet's name?

Answer

We will ask you to answer this question if you forget your password.

Save

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Force.com IDE

STONYPOINT

- Integrated Development Environment for the Force.com Platform
- Built on top of Eclipse
- Can be installed as an Eclipse plug-in or a stand alone environment
- Major Features
  - Develop Apex Code
  - Develop Visualforce pages and components
  - View and modify Metadata
  - Develop SOQL queries
  - Local copy of all Metadata
  - Can be integrated with source control

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Exercise: Install the Force.com IDE

STONYPOINT

Goal:
Install the Force.com IDE

Action Steps:

1. Navigate to: http://wiki.developerforce.com/page/Force.com_IDE_Installation
2. Install Eclipse
3. Launch Eclipse and click **Help > Install New Software...**
4. Click **Add...**.
5. In the **Add Repository** dialog, set the **Name** to "Force.com IDE" and the **Location** to http://media.developerforce.com/force-ide/eclipse42
6. Click OK
7. Check the box next to the Force.com IDE plug-in and click **Next**.
8. In the **Install Details** dialog, click **Next**.
9. In the **Review Licenses** dialog, accept the terms and click **Finish**.
10. Eclipse downloads and installs the Force.com IDE and any required dependencies. When installation is complete, you will be prompted to restart. Click **Yes**.
11. When Eclipse restarts, select **Window > Open Perspective > Other**, select **Force.com** and click **OK**. When the installation completes, launch the Force.com IDE

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Salesforce Security

STONYPOINT

- Two ways of logging in
  - Browser
    - Internet Explorer, Firefox, Chrome, etc.
  - Web Services API
    - Outlook integration, Office integration, any desktop application
- Two factor authentication
  - Salesforce standard security requires two forms of authentication
    - Something you know: Username & Password
    - Something you have: cookie, security token or Trusted IP Address
- Salesforce support Single Sign-On (SSO)

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Browser Security & Activation

STONYPOINT

- Authentication Process:
  1. User enters User Name & Password
     a. If the User Name & Password are invalid, then Salesforce will keep prompting for a valid one
  2. If valid, Salesforce checks for a cookie
     a. If a cookie exists, user logins in successfully
     b. If no cookie, Salesforce checks for a Trusted IP Address
     c. If no Trusted IP, Salesforce requires activation
        i. Salesforce sends an email to the email address associated with the User Name
        ii. User retrieves email & enters Activation Code
        iii. Salesforce puts a new Cookie on the user's machine

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Possible Causes of Activation Prompt

STONYPOINT

- Logging in from a different computer
- Logging in with a different browser on same computer
- Logging in from a different location on same computer
- Logging into a different instance of Salesforce on the same computer
- Deactivating then reactivating a User
- Clearing the cache in your browser
- Disabling cookies in your browser

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

API Security & Security Tokens

STONYPOINT

- Authentication Process:
  1. Application sends User Name & Password to API
     a. If the User Name & Password are invalid, then Salesforce will fail the login
  2. If valid, Salesforce checks for a Trusted IP Address
  3. If no Trusted IP, then Salesforce checks that a Security Token was passed
     a. If no security token was passed, login fails

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Exercise: Reset Security Token

STONYPOINT

Goal:

Reset your security token so you can use the token when creating a project using the Force.com IDE

Action Steps:

1. Navigate to [Your Name] -> My Settings
2. From the My Settings page, navigate to Personal -> Reset My Security Token
3. Follow the on screen instructions to reset your security token
4. Check your email for your new security token

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Network Security

STONYPOINT

- Trusted IP Range
  - A list of IP addresses from which Users will be able to login to Salesforce without a cookie or security token
  - Permissive, not restrictive
  - System wide
- IP Address Restrictions
  - A restricted list of IP address from which a User must be logging in from
  - Any User attempting to login from outside the range will be rejected
  - Profile based
- Login Hours
  - A range of calendar hours in which Users will be permitted to login
  - Any User attempting to login from outside the range will be rejected
  - Profile based

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Login History

STONYPOINT

- Every login attempt by every User is automatically logged by Salesforce

Login History

Login Time	Source IP	Login Type	Status	Application	Login URL
6/24/2012 4:24:16 PM PDT	97.104.82.120	Application	Success	Browser	login.salesforce.com
6/24/2012 3:40:46 PM PDT	107.20.245.152	Other Apex API	Success	N/A	
6/24/2012 11:38:07 AM PDT	107.20.245.152	Other Apex API	Success	N/A	
6/24/2012 11:03:50 AM PDT	50.16.192.154	Other Apex API	Success	N/A	
6/24/2012 11:03:48 AM PDT	50.16.192.154	Other Apex API	Success	N/A	
6/24/2012 10:55:20 AM PDT	97.104.82.120	Application	Success	Browser	login.salesforce.com
6/20/2012 7:30:27 AM PDT	50.16.192.154	Other Apex API	Success	N/A	
6/20/2012 7:30:26 AM PDT	50.16.192.154	Other Apex API	Success	N/A	
6/20/2012 7:25:19 AM PDT	50.16.192.154	Other Apex API	Success	N/A	
6/20/2012 7:25:18 AM PDT	50.16.192.154	Other Apex API	Success	N/A	

- The first place to go to troubleshoot login issues
- **Path**: Setup -> Administer-> Manage Users -> Users

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Business Scenario: Network Security

STONYPOINT

As a developer, you want to be able to login through the browser and through the API without having to provide a cookie or a security token.

It is important that you DO NOT prevent other users from accessing the system when setting up these IP addresses.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Exercise: Trusted IP Ranges

STONYPOINT

Action Steps:

1. Navigate to your user record
   a. Go to the login history related list
   b. Copy the IP address of the last successful login
2. Navigate to Setup -> Administer -> Security Controls -> Network Access
3. Create a new Trusted IP Range
   a. Paste the IP address in both the Start IP Address and End IP Address fields
   b. Save the range
4. Ask your classmates if they have a different IP address
   a. If they do, create as many ranges as needed to encapsulate all IP addresses being used in the class

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Profiles

STONYPOINT

- Profiles Control:
  - How a user sees the system
  - What a user can do within the system
    - General User Permissions
      - Ex: Run Reports, Export Reports, Convert Leads, etc.
    - Administrative Permissions
      - Ex: Customize Application, Manage Dashboards, Manage Public Reports, etc.
    - Object Permissions
      - Ex: Read, Create, Edit on Account; Read on Contract
- Users must be assigned to ONE Profile
- **Path**: Setup -> Administer -> Manage Users -> Profiles
- **Best Practice**: Always use custom profiles / NEVER use standard profiles
  - Standard Profiles cannot be customized
  - **Exception**: Use the System Administrator profile

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Profiles

STONYPOINT

- Users must be assigned to ONE Profile
- **Best Practice**: Always use custom profiles / NEVER use standard profiles
  - Standard Profiles cannot be customized
  - **Exception**: Use the System Administrator profile

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Business Scenario: Custom Profiles

STONYPOINT

In order to accomplish the objectives of expanding the system to include the ability to invoice, we need to create new Profiles. The new finance Users will need different permissions in the system. The primary way to give different Users different permissions is to assign them to different Profiles.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Exercise: Configure Profiles

STONYPOINT

Goal:

Create a new profile for the Finance users

Action Steps:

1. Navigate to Setup -> Administer -> Manage Users -> Profiles
2. Create a new profile for Finance
   a. Clone from the Standard User profile
   b. Name the Profile: GenWatt Finance
   c. Give the ability to read, create and edit Accounts
   d. Give the ability to read, create and edit Contacts
   e. Give the ability to read, create and edit Opportunities
   f. Give access to only the Sales application

Best Practice: Don't give Delete permission

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Record Ownership / Full Access

STONYPOINT

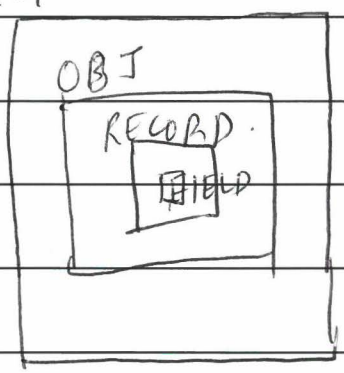
- Every record in Salesforce has an owner
- The owner of the record is given special privileges
  - The ability to delete the record
  - The ability to share the record
  - The ability to transfer ownership of the record
- These special ownership permissions are dependent on Profile permissions
- Full Access cannot be shared using Sharing

Knowledge Check:

A sales person is the owner of an Opportunity record but cannot delete it. Why might this be?

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

ORG



Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Organization Wide Defaults (OWD)

STONYPOINT

- **Path**: Setup -> Administer -> Security Controls -> Sharing Settings
- Used to set default access to records
- The only place in the application where it is possible to **restrict access** to records
- Possible Settings:
  - Private – Available for all objects
  - Public Read Only – Available for all objects
  - Public Read/Write – Available for all objects
  - Public Read/Write/Transfer - Only available for Leads and Cases
    - Gives everyone in the organization Full Access
  - Public Full Access - Only available for Campaigns
    - Gives everyone in the organization Full Access
    - Dependent on Marketing User checkbox
  - Controlled by Parent - Only available for Contacts and Activities

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Business Scenario: OWD

STONYPOINT

GenWatt is thinking about going public and has a new team of auditors helping them out. Their auditors are very concerned about the security of the GenWatt data. The new auditors have told them that Users should be given access only to the data necessary to perform their job.

Salesforce will be audited and must be in compliance with the new policies.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Exercise: Organization Wide Defaults

STONYPOINT

Action Steps:

1. Using Organization Wide Defaults, lock down Salesforce so that Users only see the records that they own. Hint: Set everything to Private.

Questions:

1. Could this objective be accomplished without using Organization Wide Defaults?
2. Is security the only good reason to restrict access to records?

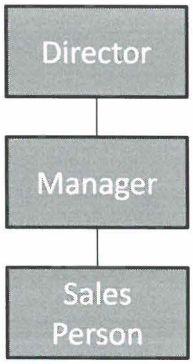
Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Role Hierarchy

STONYPOINT

- **Path**: Setup -> Administration Setup -> Manage Users -> Roles
- Used to grant access to records
- Users in Roles directly above other Roles in the chain of command are granted the same permission to records as the Role they are above



Read Only	Read/Write	Full Access
Read Only	Read/Write	Full Access
Read Only	Read/Write	Full Access

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Role Hierarchy

STONYPOINT

- Grant access using hierarchies is always the behavior for Standard objects
- Grant access using hierarchies can be turned off for Custom Objects
  - This option is available on the Organization Wide Defaults setup page
- The Role hierarchy is not necessarily the org chart
- Users can only be assigned to one Role
- Roles do not have to have Users assigned
- Opportunity and Forecast Reports allow drill down utilizing the hierarchy
  - In order to utilize this capability, the User running the report must have a Role assigned
  - The Report will automatically begin at the User's Role in the hierarchy

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Manual Sharing

STONYPOINT

- **Path**: Navigate to the record you want to share and click the Sharing button
  - Note: The Sharing button automatically disappears from the Page Layout when the OWD is set to Public Read/Write
- Used to grant additional access to an individual record
- Dependent on Profiles allowing access to Object
- Only relevant when OWD is set to Private or Public Read Only
- Can grant Read Only or Read/Write access
- Cannot grant Full Access

NOTE: One of the most common uses of Apex is to automate the manual sharing of records

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.
34

Queues

STONYPOINT

- Allows assignment of records to a Queue instead of an individual User
- Queues can have an unlimited number of members
- Queue members can be Users, Groups, Roles, Roles & Subordinates, and Partner Users
- Allows Users to be assigned to multiple Queues
- Users in Queue are granted Full Access to records owned by the Queue
- Leads, Cases and Custom Objects can be assigned to Queues
- **Path**: Setup -> Administer -> Manage Users -> Queues

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Salesforce Fields

STONYPOINT

- Fields in Salesforce exist on Standard and Custom objects
- Fields hold data
- Access to fields can be controlled using Field Level Security
- Two Categories of Fields
  - Standard Fields
    - Created automatically by Salesforce
    - Limited ability to customize
  - Custom Fields
    - Created by you
    - Highly customizable

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.
36

Salesforce Fields

STONYPOINT

- Both Standard and Custom objects have Standard fields
  - Standard fields on the standard Account object:

Account Standard Fields Account Standard Fields Help ?

Action	Field Label	Field Name	Data Type	Controlling Field
	Account Name	Name	Name	
Edit	Account Number	AccountNumber	Text(40)	
Edit	Account Owner	Owner	Lookup(User)	
Edit	Account Site	Site	Text(80)	
Replace | Edit	Account Source	AccountSource	Picklist	
Edit	Annual Revenue	AnnualRevenue	Currency(18, 0)	
	Billing Address	BillingAddress	Address	

  - Standard fields on a custom object:

Standard Fields Standard Fields Help ?

Action	Field Label	Field Name	Data Type	Controlling Field
	Created By	CreatedBy	Lookup(User)	
	Last Modified By	LastModifiedBy	Lookup(User)	
Edit	Module Name	Name	Auto Number	
Edit	Owner	Owner	Lookup(User,Queue)	

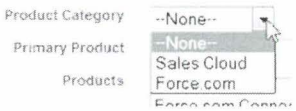
Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

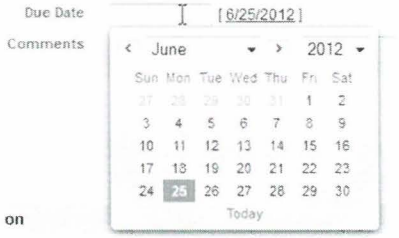
Salesforce Fields

STONYPOINT

- When defining a field in Salesforce, you are also defining the user interface element
  - Picklist field:



  - Date field:



Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Salesforce Fields

STONYPOINT

- All fields, Standard and Custom fields have the following elements
  - Name
  - Label
  - Help Text
- Custom Fields have a Description field
- Standard Fields and Fields from a Managed Package
  - Limited ability to customize
    - Change Help Text
      - We recommend you add Help Text to every field
    - Add picklist values
    - Rename Labels
  - Cannot be deleted
- Custom fields can be deleted
  - When deleted go into the deleted fields area for 15 days before being permanently deleted
  - All data in the field will also be deleted

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Custom Fields

STONYPOINT

Auto Number	A system-generated sequence number that uses a display format you define. The number is automatically incremented for each new record.
Formula	A read-only field that derives its value from a formula expression you define. The formula field is updated when any of the source fields change.
Roll-Up Summary *	A read-only field that displays the sum, minimum, or maximum value of a field in a related list or the record count of all records listed in a related list.
Lookup Relationship	Creates a relationship that links this object to another object. The relationship field allows users to click on a lookup icon to select a value from a popup list. The other object is the source of the values in the list.
Master-Detail Relationship	Creates a special type of parent-child relationship between this object (the child, or "detail") and another object (the parent, or "master"), where: • The relationship field is required on all detail records. • The ownership and sharing of a detail record are determined by the master record. • When a user deletes the master record, all detail records are deleted. • You can create rollup summary fields on the master record to summarize the detail records. The relationship field allows users to click on a lookup icon to select a value from a popup list. The master object is the source of the values in the list.
Checkbox	Allows users to select a True (checked) or False (unchecked) value.
Currency	Allows users to enter a dollar or other currency amount and automatically formats the field as a currency amount. This can be useful if you export data to Excel or another spreadsheet.
Date	Allows users to enter a date or pick a date from a popup calendar.
Date/Time	Allows users to enter a date and time, or pick a date from a popup calendar. When users click a date in the popup, that date and the current time are entered into the Date/Time field.
Email	Allows users to enter an email address, which is validated to ensure proper format. If this field is specified for a contact or lead, users can choose the address when clicking Send an Email. Note that custom email addresses cannot be used for mass emails.
Number	Allows users to enter any number. Leading zeros are removed.
Percent	Allows users to enter a percentage number, for example, 10, and automatically adds the percent sign to the number.
Phone	Allows users to enter any phone number. Automatically formats it as a phone number.
Picklist	Allows users to select a value from a list you define.
Picklist (Multi-Select)	Allows users to select multiple values from a list you define.
Text	Allows users to enter any combination of letters and numbers.
Text Area	Allows users to enter up to 255 characters on separate lines.
Text Area (Long)	Allows users to enter up to 32,768 characters on separate lines.
Text Area (Rich)	Allows users to enter formatted text, add images and links. Up to 32,768 characters on separate lines.
Text (Encrypted)	Allows users to enter any combination of letters and numbers and store them in encrypted form.
URL	Allows users to enter any valid website address. When users click on the field, the URL will open in a separate browser window.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

40

Pick Lists

STONYPOINT

- Two types: Pick list & Pick list (Multi-Select)
- Allow you to add, delete and find & replace values
- If you remove a value from a pick list field, it does NOT remove it from all existing records
  - You must do a Replace if you want that to occur
- A default value can be selected
- Values can be sorted alphabetically
  - Numbers sort alphabetically as well
    - 1,10,2,20,3

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Formula Fields

STONYPOINT

- Calculated fields
- Use very similar syntax to Excel
- Are always calculated at run time
  - The value of the formula is never stored in the database
- Allow calculations across related objects
  - Can include up to 10 related objects across all formula fields
  - The are known as "Cross Object" formulas
- Can include special content like Images and Hyperlinks
- Great for displaying extra data on related lists or email templates

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Field Level Security (FLS)

STONYPOINT

- Sets permission field by field by Profile
- Choices are read/write, read only or hidden
- All or none - not conditional
- At database Model level
- Dependent on Object level security
  - If no access to Object, FLS is irrelevant
- Can be configured & viewed in 3 places:
  - Profile, Field, Field Accessibility
- Formula Fields, Workflow and Apex ignore field level security
  - Visualforce pages enforce field level security

Best Practice: Use field level security to truly secure fields

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Security Conflicts

STONYPOINT

- The table below explains the rules applied when a conflict occurs between two security settings

First Setting	Second Setting	Access Level
Profile Setting	Permission Set Setting	Highest Level Granted
Sharing Setting	Sharing Setting	Highest Level Granted
Sharing Setting	Profile / Permission Set Setting	Lowest Level Granted
Profile Setting	Field Level Security Setting	Lowest Level Granted
Sharing Setting	Field Level Security Setting	Lowest Level Granted

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Object Relationships

STONYPOINT

- Object can be related to one another
- Relationships are created when a relationship field is added to an object
- Three types of relationship fields:
  - Lookup
  - Master Detail
  - Hierarchical (User Object only)
- The object with the lookup field on it is referred to as the child in the relationship
- The object being related is referred to as the parent in the relationship
- Lookup and Master Detail fields have similar but different characteristics

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Lookup vs. Master Detail

STONYPOINT

Lookup Fields	Master Detail Fields
Up to 40 Lookup fields allowed per object	A maximum of 2 Master Detail fields per object
Lookup fields are not required	Master Detail fields are always required
When a parent is deleted, the child is not affected	When a parent is deleted, all child records are deleted
Object has an owner field	Object does not have an owner field
Does not allow Rollup Summary fields	Allows Rollup Summary fields
Child has independent sharing and security	Child inherits all sharing and security from parent record

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Determining Object Relationships

STONYPOINT

- Fields
  - You can determine the relationships between objects by navigating the object customization screen and looking at all the field definitions
  - Master Detail relationships will be shown in the main list of Objects

Action	Label	Installed Package	Master Object	Deployed	Description
Edit	Activity	LMS ILT	Development Plan	✓	
Edit	Assignment	LMS Learning	Transcript, Training User License	✓	LMS Consumer Application
Edit	Cancellation Policy	LMS ILT		✓	
Edit	Cart	LMS Learning		✓	LMS Consumer Application
Edit	Catalog	LMS Learning		✓	
Edit	Catalog Account	LMS Learning	Account, Catalog	✓	

- You can also utilize the Schema Builder
  - Allows you to create a visual diagram of object relationships
  - Allows you to edit objects

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Exercise: Schema Builder

STONYPOINT

Action Steps:

1. Navigate to the Schema Builder. Path: Setup -> Build -> Schema Builder
2. From the menu on the left, make sure the Objects tab is selected
3. From the drop down list named Select from, make sure All Objects is selected
4. Check the Clear All link to deselect all objects and clear the screen
5. Add the following Objects to the Schema Builder diagram by checking the box next to the Object:
   a. Account
   b. Case
   c. Contact
   d. Opportunity
   e. Opportunity Product
6. When you feel like you understand the relationships between these Objects, click the Close button to close the Schema Builder
   a. You cannot save diagrams

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Rollup Summary Fields

STONYPOINT

- Calculate values from a set of child records and display the result on the parent record
- Rollup Summary fields can perform the following calculations:
  - Max
  - Min
  - Sum
  - Count
- Records included in the calculation can be filtered
- On standard objects, Rollup Summary fields are only available on Accounts, Opportunities & Campaigns
- On custom objects, the child records must be directly related to the parent through a master-detail relationship
- Unlike Formula Fields, values are stored on the parent record in the database
  - Modification of a child record causes a save event on the parent record
  - This causes Workflow, Validation rules & Apex triggers on the parent record to fire

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Business Scenario: Rollup Summary Fields

STONYPOINT

GenWatt wants to keep track of their most valuable customers and their most promising prospects. When they look at an Account record, they want to know two things: the total amount of all Won opportunities and the total amount of all Open opportunities.

To accomplish this, create two new rollup summary fields on the Account object.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Exercise: Rollup Summary Fields

STONYPOINT

Goal:

Create a new Custom Field on the Account to calculate the amount of all won opportunities

Action Steps:

1. Navigate to Setup -> Build -> Customize-> Accounts -> Fields
2. Create a new custom Field of Type Rollup Summary
3. Choose the following options
   a. Field Label: Won Opportunities
   b. Field Name: Won_Opportunities
   c. Summarized Object: Opportunities
   d. Rollup Type: SUM
   e. Field to Aggregate: Amount
   f. Filter Criteria: Only records meeting certain criteria should be included in the calculation
   g. Filter: Won equals True
4. Make the field visible to all Profiles
5. Add the field to all the Page Layouts
6. Save the field

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Exercise: Rollup Summary Fields

STONYPOINT

Goal:

Create a new Custom Field on the Account to calculate the amount of all open opportunities

Action Steps:

1. Navigate to Setup -> Build -> Customize-> Accounts -> Fields
2. Create a new custom Field of Type Rollup Summary
3. Choose the following options
   a. Field Label: Open Opportunities
   b. Field Name: Open_Opportunities
   c. Summarized Object: Opportunities
   d. Rollup Type: SUM
   e. Field to Aggregate: Amount
   f. Filter Criteria: Only records meeting certain criteria should be included in the calculation
   g. Filter: Closed equals False
4. Make the field visible to all Profiles
5. Add the field to all the Page Layouts
6. Save the field

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

52

Custom Objects

STONYPOINT

- Custom objects store information for unique, non-CRM business processes
- Have standard fields
  - Created By, Created Date, Last Modified By, Last Modified Date, Id, Owner
  - Name
    - Can either be text or auto-number
- Can optionally
  - Allow Reports
  - Allow Activities
  - Track Field History
- Can use Visualforce for custom help page
- Deployment Status
  - If not deployed, only System Administrator can see object and records in the object

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Business Scenario: Custom Objects

STONYPOINT

GenWatt wants to begin generating and tracking invoices within Salesforce. Whenever an Opportunity record is set to Closed – Won, a new Invoice record should be created.

GenWatt management has decided that only certain people should be able to see invoices. The invoice should be visible to the sales person who owned the Opportunity and the SVP of Customer Service. The SVP of Custom service should not be able to see the Opportunity record associated with the invoice, just the invoice.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Exercise: Invoice Custom Object

STONYPOINT

Goal:

Create a new custom object for Invoices.

Action Steps:

1. Navigate Setup -> Build -> Create -> Objects
2. Create a new custom object by clicking on the New Custom Object button
3. Set the following options:
   a. Label: Invoice
   b. Plural Label: Invoices
   c. Object Name: Invoice
   d. Record Name: Invoice #
   e. Data Type: Auto Number
   f. Display Format: INV-{000000}
   g. Starting Number: 1001
   h. Allow Reports
   i. Deployed
4. Save the object by clicking the Save button.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

55

Exercise: Custom Tabs

STONYPOINT

Goal:

Create a Custom Tab for the Invoice object

Action Steps:

1. Navigate to Setup -> Build -> Create -> Tabs
2. Create a new Custom Object Tab with the following settings:
   a. Object: Invoice
   b. Tab Style: Your Choice of icon and colors
3. Set the Tab visibility to:
   a. System Administrator, Custom: Support Profile, Custom: Sales Profile: Default On
   b. All other Profiles: Default Off
4. Add the Tab to the Sales App and the Call Center App

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Exercise: Invoice Custom Fields

STONYPOINT

Action Steps:

1. Navigate to Setup -> Build -> Create -> Objects
2. Click on the Invoice Object and navigate to the Custom Fields & Relationships section. Create the following new fields:

Type	Label	Name	Details
Lookup	Opportunity	Opportunity	Related To: Opportunity Don't allow deletion of the lookup record that's part of a lookup relationship. Make Visible to all Profiles Add related list to all Opportunity Page Layouts
Lookup	Account	Account	Related To: Account Always require a value in this field in order to save a record Make Visible to all Profiles Add related list to all Account Page Layouts
Date	Due Date	Due_Date	Make Visible to all Profiles Add to Invoice Page Layout

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Exercise: Invoice Custom Fields

STONYPOINT

Type	Label	Name	Details
Picklist	Status	Status	Values: Due (default), Overdue, Paid Make Visible to all Profiles Add to Invoice Page Layout
Currency	Amount	Amount	Length: 16 Decimal Places: 2 Required Make Visible to all Profiles Add to Invoice Page Layout
Currency	Amount Paid	Amount_Paid	Length: 16 Decimal Places: 2 Make Visible to all Profiles Add to Invoice Page Layout
Formula	Balance	Balance	Return Type: Currency, 2 Decimal Places Formula: Amount__c - Amount_Paid__c Make Visible to all Profiles Add to Invoice Page Layout

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Exercise: Invoice Custom Fields

STONYPOINT

Type	Label	Name	Details
Lookup	Billing Contact	Billing_Contact	Child Relationship Name: Invoices Don't allow deletion of the lookup record that's part of a lookup relationship Make Visible to all Profiles Add to Invoice Page Layout Related List Label: Invoices Add to all the Contact page layouts

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

List Views

STONYPOINT

- Available on most tabs
- Show only records of one type of object
- Are customizable
  - Filters
  - Columns
- Provide nice navigation features
  - Edit returns to list
  - View provides a breadcrumb back to list
- Can be built by end users
- Can be secured individually

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Exercise: List Views

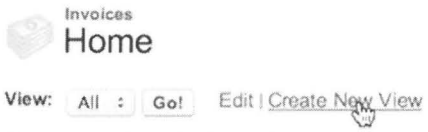
STONYPOINT

Goal:

Create a new List View for Invoices

Action Steps:

1. Navigate to the Sales application and then to the Invoices tab
2. Create a new view by clicking on the Create New View link



3. Choose the following options
   a. View Name: All Invoices
   b. View Unique Name: All_Invoices
   c. Filter By Owner: All Invoices
   d. Select whatever you would like to display in the Select Fields to Display area
4. Make the View visible to all users in the Restrict Visibility area
5. Save the View

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Force.com IDE Project

STONYPOINT

- Contains selected Metadata from a Salesforce organization
  - Typically items included:
    - Apex Classes & Triggers
    - Visualforce Pages & Components
    - Standard & Custom Objects
  - Most other Metadata items can be included
- Creates a local copy of all selected Metadata

Best Practices:

- Include only the Metadata items you intend to work with
- Use source control

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Exercise: Create a Project

STONYPOINT

Goal:

Create a Force.com project using the Force.com IDE

Action Steps:

1. From within the Force.com IDE, navigate to File -> New -> Force.com Project
2. Name the project SPDEV-205 and login using the credentials you created for this class
3. When prompted to choose the project contents, choose Apex and Visualforce
4. After making your selection, your project will be created

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

SOQL & SOSL

STONYPOINT

- SOQL (Salesforce Object Query Language)
  - Very similar to SQL
  - Query only
    - Cannot be used to modify or create data
    - Cannot be used to modify object structures
  - No ad hoc table joins allowed
    - All joins based on pre-defined object relationships
  - Case Insensitive
- SOSL (Salesforce Object Search Language)
  - Duplicates the functionality of the global search
  - Can search for the search phrase across multiple objects at once
  - Case Insensitive

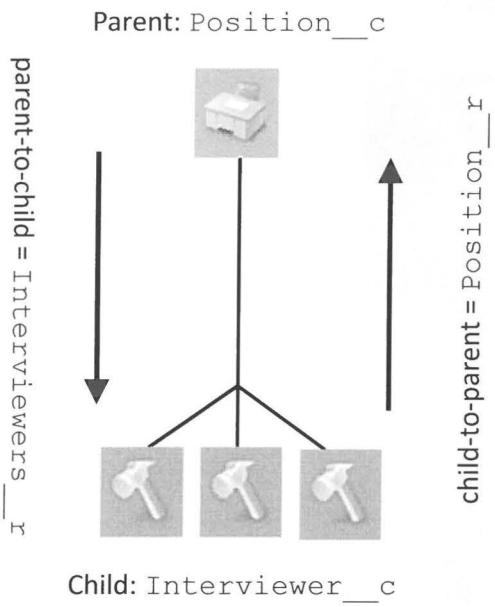
Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

sObject Relationships

STONYPOINT

- Standard relationship
  - Created by Salesforce.com and cannot be modified
- Custom relationship
  - Created by a developer
  - Can be modified by a developer
  - Both directions are appended with `__r`
- A relationship has different names depending on the direction you are traversing:
  - Parent-to-child: plural version of the child object.
  - Child-to-parent: singular version of the parent object.



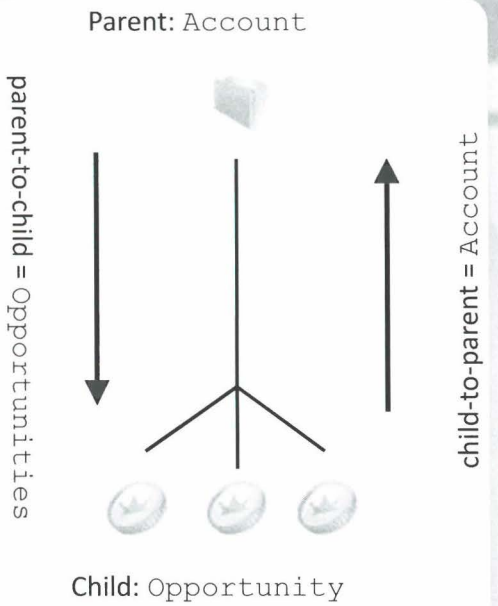
Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

sObject Relationships in Code

STONYPOINT

- Relationships in Apex and SOQL are written using dot notation
- Allows a developer to query multiple tables without an explicit table join
- Can traverse up to five levels parents
- Can include only one level of children




```
Opportunity.Account.Industry = 'Mining'
```

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

SOQL Queries: Right Outer Join

STONYPOINT

Position__c

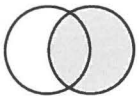
Id	Name	Dept__c
a05S0000000WZeY	Programmer	Engineering
a05300000050Krc	Account Executive	Sales
a05S0000000X2eY	Billing Assistant	Finance
a05300000050KrT	Product Manager	Engineering

Job_Application__c

Name	Position__c	Status__c
APP-0069	a05S0000000WZeY	Open
APP-0070		Open
APP-0071	a05S0000000WZeY	Closed
APP-0072	a05300000050Krc	Open

Get job applications with associated position data

```
SELECT Name, Position__r.Dept__c  FROM Job_Application__c
```



List<Job_Application__c>

Name	Position__r.Dept__c
APP-0069	Engineering
APP-0070	
APP-0071	Engineering
APP-0072	Sales

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

SOQL Queries: Left Outer Join

STONYPOINT

Position__c

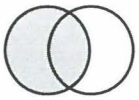
Id	Name	Dept__c
a05S0000000WZeY	Programmer	Engineering
a05300000050Krc	Account Executive	Sales
a05S0000000X2eY	Billing Assistant	Finance
a05300000050KrT	Product Manager	Engineering

Job_Application__c

Name	Position__c	Status__c
APP-0069	a05S0000000WZeY	Open
APP-0070		Open
APP-0071	a05S0000000WZeY	Closed
APP-0072	a05300000050Krc	Open

Get positions and related job application data

```
SELECT Name, (SELECT Name FROM Job_Applications__r)
FROM Position__c
```



List<Position__c>

Name	Job_Applications__r
Programmer	(Related List)
Billing Assistant	
Account Executive	(Related List)
Product Manager	

List<Job_Application__c>

Name
APP-0069
APP-0071

Name
APP-0072

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

SOQL Queries: Semi-Join (Left Inner Join)

STONYPOINT

Position__c

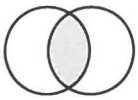
Id	Name	Dept__c
a05S0000000WZeY	Programmer	Engineering
a05300000050Krc	Account Executive	Sales
a05S0000000X2eY	Billing Assistant	Finance
a05300000050KrT	Product Manager	Engineering

Job_Application__c

Name	Position__c	Status__c
APP-0069	a05S0000000WZeY	Open
APP-0070		Open
APP-0071	a05S0000000WZeY	Closed
APP-0072	a05300000050Krc	Open

Only get positions with related job applications

```
SELECT Name FROM Position__c
WHERE Id IN (SELECT Position__c FROM Job_Application__c)
```



List<Position__c>

Name
Programmer
Account Executive

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

SOQL Queries: Right Inner Join

STONYPOINT

Position__c

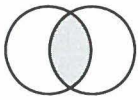
Id	Name	Dept__c
a05S0000000WZeY	Programmer	Engineering
a05300000050Krc	Account Executive	Sales
a05S0000000X2eY	Billing Assistant	Finance
a05300000050KrT	Product Manager	Engineering

Job_Application__c

Name	Position__c	Status__c
APP-0069	a05S0000000WZeY	Open
APP-0070		Open
APP-0071	a05S0000000WZeY	Closed
APP-0072	a05300000050Krc	Open

Only get job applications with related positions

```
SELECT Name, Status__c, Position__r.Name
FROM Job_Application__c WHERE Position__c != null
```



List<Job_Application__c>

Name	Status__c	Position__r.Name
APP-0069	Open	Programmer
APP-0071	Closed	Programmer
APP-0072	Open	Account Executive

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

SOQL Queries: Right Anti-Join

STONYPOINT

Position__c

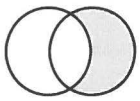
Id	Name	Dept__c
a05S0000000WZeY	Programmer	Engineering
a05300000050Krc	Account Executive	Sales
a05S0000000X2eY	Billing Assistant	Finance
a05300000050KrT	Product Manager	Engineering

Job_Application__c

Name	Position__c	Status__c
APP-0069	a05S0000000WZeY	Open
APP-0070		Open
APP-0071	a05S0000000WZeY	Closed
APP-0072	a05300000050Krc	Open

Only get job applications without related positions

```
SELECT Name, Status__c FROM Job_Application__c
WHERE Position__c = null
```



List<Job_Application__c>

Name	Status__c
APP-0070	Open

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

SOQL Queries: Left Anti-Join

STONYPOINT

Position__c

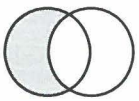
Id	Name	Dept__c
a05S0000000WZeY	Programmer	Engineering
a05300000050Krc	Account Executive	Sales
a05S0000000X2eY	Billing Assistant	Finance
a05300000050KrT	Product Manager	Engineering

Job_Application__c

Name	Position__c	Status__c
APP-0069	a05S0000000WZeY	Open
APP-0070		Open
APP-0071	a05S0000000WZeY	Closed
APP-0072	a05300000050Krc	Open

Only get positions without related job applications

```
SELECT Name FROM Position__c WHERE Id
NOT IN (SELECT Position__c FROM Job_Application__c)
```



List<Position__c>

Name
Billing Assistant
Product Manager

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Multi-Level Relationships

STONYPOINT

Relationship names can be chained together to access parent or child records

- Up to five levels deep for child-to-parent relationships.
- One level deep parent-to-child relationship.

Parent Record Example:

```
SELECT o.Opportunity.Account.Name
FROM OpportunityLineItem o
```

Children Records Example:

```
SELECT o.Name,
(Select PricebookEntryId FROM
OpportunityLineItems) FROM Opportunity o
```

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Important SOQL Keywords

STONYPOINT

- ORDER BY
  - ASC or DESC
- LIMIT
  - Limits the records returned
  - When used with ORDER BY, sorts first then applies limit
- GROUP BY
  - ROLLUP to calculate subtotals
- IN
- LIKE
  - % is wildcard
- FOR UPDATE
  - Locks the returned record from being updated by another request
- ALL ROWS
  - Returns both active records and records in the Recycle Bin

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Exercise: Force.com SOQL Reference Guide

STONYPOINT

Goal:

Create a bookmark to the Force.com SOQL and SOSL Reference

Action Steps:

1. In a new browser window, navigate to http://developer.force.com
2. In the Search bar, type in SOQL Reference.
3. The first search result should be the guide.
4. Click on the link and create a bookmark to the page that comes up

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Walkthrough: SOQL Queries

STONYPOINT

Goal:

Use the Force.com IDE to perform the queries below

Action Steps:

1. Select all the Accounts and any related Opportunities
2. Select all Accounts and all related Cases and all related Contacts
3. Select all Cases, the name of the Account associated with the Case and the Name of the Contact associated with the Case
4. Select all Products and any Opportunities they are related to

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

1) Select Name (select Name from Opp) from Account.

2) Select Name, (Select Name from Contacts), (Select CaseNumber from Cases) from Account.

3)

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Force.com Architecture

STONYPOINT

- Salesforce separates it's architecture into two distinct development methods: Declarative and Programmatic
- Declarative
  - Clicks not code
  - The preferred method of development
  - Automatically upgraded
  - Examples: Page Layouts, Validation Rules, Workflow Rules, Formula Fields
- Programmatic
  - Code
  - Apex
  - Web Services API using standard tools
  - Not automatically upgraded
    - Locked to a specific version of the API
  - Should only be used as a last resort

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Model – View – Controller (MVC)

STONYPOINT

- The idea that software development can be separated into three distinct functional areas
- Academic in nature and not practical in reality
- Salesforce attempts to support this paradigm
- **Model**
  - The database layer
  - Contains objects, fields, and security
- **Controller**
  - The business logic layer
  - Links the Model to the View
  - Contains Workflow, Validation Rules, Apex
- **View**
  - The user interface layer
  - Tabs, Page Layouts, Visualforce

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Force.com Architecture & MVC

STONYPOINT

	Declarative	Programmatic
View	Page Layouts Tabs List Views Visual Workflow	Visualforce External Program
Controller	Validation Rules Workflow Rules Approval Processes Assignment Rules Auto-Response Rules Escalation Rules	Apex Triggers Apex Classes Web Services API Integration
Model	Objects Profiles	Objects Profiles

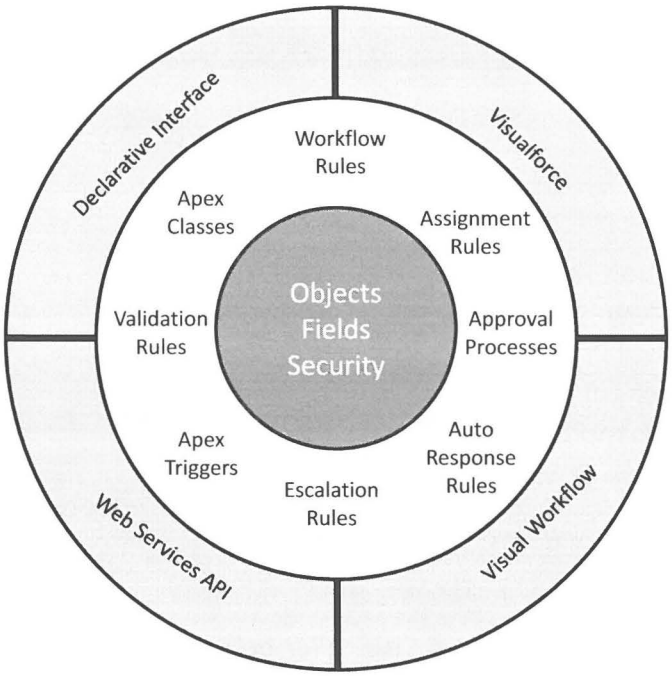
On the Force.com Platform, the Model & Controller behave as a single unit

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Model - View - Controller

STONYPOINT



Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

80

What is Automation?

STONYPOINT

- Any technology within Salesforce that is used to simplify utilization of the system for the end users.
- Salesforce considers the following technologies to be automation:
  - Rollup Summary Fields
  - Validation Rules
  - Workflow Rules & Approval Processes
  - Assignment Rules & Auto-Response Rules
  - Escalation Rules
  - Scheduling of Reports & Dashboards
  - Apex & Visualforce
  - Visual Workflow (Force.com Flow)

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Data Validation

STONYPOINT

- Standard Validation
  - Built into system
  - Examples:
    - Date field must contain a date
    - Picklist fields
    - Lookup fields
    - Number field must contain numbers
    - Field is required at the object level
- Custom Validation
  - Used to enforce business rules
  - Examples:
    - Validation Rules
    - Apex Triggers

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Validation Rules

STONYPOINT

- Enforce business rules
- Often used to:
  - Enforce conditionally required fields
  - Enforce special formatting
  - Prevent modifications of records
- Do not require writing of code
- Use formula syntax to test for error condition
  - Tests that the error condition is true, not that the data is good
- Has two parts:
  - Error condition
  - Error message
    - Can be displayed next to a specific field or at the top of the page
- Execute when a record is saved
  - Don't execute on delete
  - Don't affect existing records until they are modified
- Must be activated to work

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Business Scenario: Validation Rules

STONYPOINT

GenWatt would like to enforce the following additional business logic:

- If an Opportunity is Open, it must have a Close Date no earlier than today

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Exercise: Validation Rules

STONYPOINT

Goal:

Create a new Validation Rule on the Opportunity object to force a Close Date in the future

Action Steps:

1. Navigate to Setup -> Build -> Customize-> Opportunity -> Validation Rules
2. Create a new Validation rule and choose the following options:
   a. Rule Name: Force_Future_Close_Date
   b. Active: Checked
   c. Formula: NOT(IsClosed) && CloseDate < TODAY()
   d. Error Message: You cannot save an open opportunity with a Close Date in the past
   e. Error Location: Field: Stage
3. Save the rule

Best Practice: Don't put error messages on Date fields

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Workflow

STONYPOINT

- Used to automate certain things within Salesforce
- Can perform the following 4 actions
  - Email Alert
  - Task
  - Field Update
  - Outbound Message
- Can perform any of these actions either immediately or at some point in the future
- Each Workflow Rule is tied to one object
  - It is not possible to create a Workflow Rule that works for more than one object
- Workflow fires when a record is saved
  - If you create a Workflow rule, it will not affect existing records until someone tries to save the record again
- When a record is saved, criteria is evaluated on each Workflow rule and the Workflow only fires if the criteria evaluates to true

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Workflow Actions

STONYPOINT

- Email Alert
  - Sends an Email Template to Leads, Contacts or Users that are somehow related to the record causing the Workflow to fire
  - Can also send email to any Salesforce User
  - Can also send to an email address in an Email field on the record causing the Workflow to fire
  - Can also send to an additional 5 hard coded email addresses
    - These email addresses do not need to be Salesforce Users
- Task
  - Assigns a Task to a User or Role
  - Can also notify the Assignee via email when the Task is assigned
  - The Task is automatically associated with the record that triggered the Workflow

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Workflow Actions

STONYPOINT

- Field Update
  - Update the value of a field on the record that triggers the Workflow to fire
  - Cannot create new records
  - Cannot delete records
  - Cannot modify related records
    - Exception: If the record triggering the Workflow rule is a child in a Master Detail relationship, the Field Update can update a field on the parent record
- Outbound Message
  - Sends an XML message to an external web service
  - Used for integration with external systems

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Business Scenario: Workflow

STONYPOINT

GenWatt wants to improve their pipeline management. They want to ensure that sales people follow a naming convention and keep Opportunity records accurate. GenWatt has decided that trying to manually enforce a naming convention isn't working so they want an automatic naming convention enforce.

This can be accomplished with Workflow.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Exercise: Workflow Field Update

STONYPOINT

Goal:

1. Create a new Workflow Rule that will use a Field Update to update the Name field on the Opportunity object

Action Steps:

1. Navigate to Setup -> Build -> Create -> Workflow & Approvals -> Workflow Rules
2. Create a new Workflow Rule
3. Select the Opportunity object as the object type
4. Set the following options
   a. Rule Name: Set Opportunity Name
   b. Evaluate Rule: Every time a record is created or edited
   c. Rule Criteria: formula evaluates to true
   d. Enter the following formula into the formula field:
      1=1
5. Save the criteria by clicking the Save & Next button

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Exercise: Workflow Field Update

STONYPOINT

Action Steps:

6. Add an immediate Workflow Action by clicking the Add Workflow Action button in the Immediate Workflow Actions section
   a. Add a New Field Update and set the following options:
      i. Name: Set Opportunity Name
      ii. Unique Name: Set_Opportunity_Name
      iii. Field to Update: Opportunity Name
      iv. Use a formula to set the new value and enter the following formula
         Account.Name & " - " & TEXT(Type) & " - " & TEXT(MONTH(CloseDate)) & "/" & TEXT(YEAR(CloseDate))
      v. Save the Field Update
7. Click the Done button to finish the rule
8. Activate the Rule by clicking the Activate button
9. Test the rule by opening any Opportunity record and saving it
   a. The name should change to reflect the new naming convention

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Email Templates

STONYPOINT

- Used to send email from within Salesforce
- Have similar behavior to Mail Merge in that fields are created in the template that are replaced with data when sent
- Can be sent:
  - Manually, one at a time, using the user interface
  - Using mass email
  - Automatically through Workflow
- Four Types:
  - Text
  - HTML (using Letterhead)
  - Custom (without using Letterhead)
  - Visualforce
- Current Stony Point recommendation is to use Text templates
  - Text works the best on mobile devices
- Path: Setup -> Administer -> Communication Templates -> Email Templates

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Salesforce Record Id's

STONYPOINT

- Every record in Salesforce has an ID
  - Includes data objects and configuration objects
- The first 3 characters of an ID identifies the object
  - Can be used as a programming shortcut
- ID's can be either:
  - 15 character, case sensitive
    - Used / retrieved in browser from
      - URL
        - » Can navigate directly to a record by entering the ID in URL
        - » /e enter edit mode
      - Reports
      - Lists
      - Formula Fields
  - 18 character, case insensitive
    - Retrieved via API
    - API can accepted either 15 or 18 character as input

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Development Steps

STONYPOINT

1. Declarative
2. Apex
3. Visualforce
4. Heroku & Force.com Canvas
5. 3rd party

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Apex

STONYPOINT

- A proprietary development language created by salesforce.com
  - Syntax for language is a combination of Java and C#
  - Only runs on the Force.com platform
- Not another name for Visualforce, a different technology
- Supports most object oriented constructs
- Very limited in scope
  - Primary functionality available is math, logic and data manipulation
- Fits into the Controller level of MVC
  - No ability to interact directly with the screen
  - Can be used in conjunction with Visualforce as a custom controller for a Visualforce page

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Apex

STONYPOINT

- A Java or C# development skill set makes the easiest transition to Apex development
  - Any developer with object oriented programming experience should be able to pick it up in 1-2 months
- Administrators typically cannot make the leap to writing Apex
- Force.com IDE is the development environment of choice
- Can be used to create:
  - Triggers
  - Classes
  - Custom Web Services
  - Custom Email Handlers
  - Batch Process

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

When to use Apex

STONYPOINT

- To update fields on a related record
- To create new records
- To delete records
- To update Lookup fields on a record
- To automate manually sharing of records
- Complex validation rules
- To update records with values from another record
- To create custom behavior
- To call an external web service synchronously
- When you can't do it declaratively!

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

How to Deploy Apex

STONYPOINT

- Apex can be deployed as either a class or a trigger
- Triggers
  - More procedural than object oriented
  - Bound to specific data events on a specific object
  - Code is invoked when a data event fires
- Classes
  - Support most object oriented constructs
  - Must be invoked by another process

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Apex Key Concepts

STONYPOINT

- Bulk Processing
  - Triggers are written to process records in bulk / batch
  - Instead of firing one time per record, the trigger needs to be coded to fire one time per every two hundred records
  - "Bulkify" – writing a trigger to properly support multiple records
- Limits
  - Because Salesforce is a multi-tenant environment, you are limited in the volume of resources you can consume
- Unit Tests
  - All Apex code must be covered by Unit Tests
  - 75% of all code must be covered
  - Every trigger must have at least 1% test coverage
- With Sharing / Without Sharing
  - Apex code can be configured to either respect or ignore security
  - Default behavior is to ignore security

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Limits

STONYPOINT

- Force.com has two types of limits
  - Cumulative Limits
  - Governor Limits
- Cumulative Limits
  - Salesforce keeps track of limits over a rolling 24 hours
  - Examples:
    - Total # of inbound API Calls per day
    - Total # of mass email messages per day - 5,000
    - Total # of web to lead Leads per day
- Governor Limits
  - Per transaction and reset after each transaction
  - No documented total limit
  - Examples:
    - A SOQL query can return no more than 50,000 rows
    - A DML statement can process no more than 10,000 rows
    - The total heap size of a transaction is limited to 6 MB

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Exercise: Force.com Apex Code Developer's Guide

STONYPOINT

Goal:

Create a bookmark to the Force.com Apex Code Developer's Guide

Action Steps:

1. In a new browser window, navigate to http://developer.force.com
2. In the Search bar, type in Apex Code Developer's Guide.
3. The first search result should be the guide.
4. Click on the link and create a bookmark to the page that comes up

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Apex Data Types

STONYPOINT

- Apex is strongly typed
- All variable types are objects
- Major types
  - Primitives (`Integer, Date, DateTime, Long, String, Boolean…`)
  - sObjects (`sObject`, or `Account`, `Position__c`, …)
  - Collections (`List, Set, or Map`)
  - An object created from user- or system-defined classes
  - Null (for the `null` constant)
  - `Object`

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Primitive Data Types

STONYPOINT

Apex uses the primitive data types supported by SOAP

- Boolean
- Date
- Datetime
- Time
- ID (18 character, although can accept 15-digit and convert it)
- Integer, Long, Double, Decimal
- String
- Blob

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

sObject Data Type

STONYPOINT

- Standard or Custom Objects as defined in the declarative interface
  - Custom objects always end in __c
    - Example: Invoice__c
- Characteristics
  - Hold Data
    - Can be returned from a SOQL query
    - Can be created in memory programmatically
  - Respect security
  - Have Fields
    - Always have an ID Field
      - Cannot be set programmatically
      - If the ID field is not null, it indicates the record already exists in the database
    - Custom fields always end in __c
      - Example: Invoice__c.Amount__c

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Apex & Metadata

STONYPOINT

- Apex is strongly typed to Salesforce metadata.
  - All sObjects and fields can be accessed using Apex.
- **Apex is Metadata aware.**
  - Declarative features referenced by Apex are protected from changes
    - Example: An administrator will not be able to delete a field that is referenced within Apex code
  - Pro: It greatly reduces the likelihood of code being broken
  - Con: It makes it difficult to remove unwanted functionality after it has been deployed
- Upgrades
  - Apex is not auto-upgraded with each new release
  - Apex is saved with a specific version of the API and the functionality of each version of the API is preserved with each release

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Apex Collections

STONYPOINT

- Set
  - An unordered list of unique keys
- List
  - An ordered list of objects
  - Can only sort ascending
  - Can be nested
- Map
  - A key, value pair
  - Essentially a set and a list
  - Can be nested
  - Used primarily for in-memory table joins

Copyright © 2012 - 2013 Stony Point, Inc. All rights reserved.

Copyright © 2012 Stony Point, Inc. All rights reserved.

Looping

STONYPOINT

- Apex supports five types of loops:
  - `do{statement} while(Boolean_condition);`
  - `while(Boolean_condition) statement;`
  - `FOR(initialization; Boolean_exit_condition; increment) statement;`
  - `FOR(variable : list_or_set) statement;`
  - `FOR(variable : [inline_soql_query]) statement;`
- All loops support following methods:
  - `break`: exits the loop fully
  - `continue`: exits the current iteration and moves to the next iteration of the loop

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.
107

Do-While & While Loops

STONYPOINT

- The `do-while` loop repeatedly executes as long a particular Boolean condition remains true.

```
Integer count = 1;
do{
  System.debug(count);
  count++;
} while(count < 11);
```

- The `while` loop checks the condition before the first iteration.

```
Integer count = 1;
while(count < 11) {
  System.debug(count);
  count++;
}
```

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

108

For Loops

STONYPOINT

- Iterating with an integer

```
1  FOR(Integer i = 0; i < 10; i++) {
2    System.debug(i+1);
3  }
```

- Iterating through a List

```
1  List<Integer> myInts = new List<Integer>();
2  myInts.add(1);
3  FOR(Integer i : myInts) {
4    System.debug(i);
5  }
```

- Iterating through a query result

```
1  FOR(Account a : [Select Name from Account]) {
2    System.debug(a.Name);
3  }
```

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

SOQL Returned Data

STONYPOINT

Results are returned as a `List`

- When using Group By, results are returned as an `AggregrateResult`

Only fields that are selected are populated

- Unpopulated fields will return `null`.
  - Attempts to reference fields not explicitly queried will result in a run-time exception
- Exception: The `ID` field is implicitly returned.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Writing SOQL in Apex

STONYPOINT

- There are two ways to embed SOQL into Apex code
- Square-bracketed expressions

```
List<Account> accounts = [SELECT Name FROM Account];
```

- String expressions

```
String s = 'SELECT Name FROM ACCOUNT';
List<Account> accounts = Database.query(s);
```

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Looping Through Query Results

STONYPOINT

There are three ways to retrieve database query results:

1. Select directly into a list (OK, but not great)

```
List<Account> accounts = [SELECT Name FROM Account];
```

2. Select directly into a loop and build a list (Very good)

```
List<Account> accounts = new List<Account>();
for (Account a : [SELECT Name FROM Account]) {
        accounts.add(a);
}
```

3. Select into a List and iterate (Very good, required for batch Apex)

```
List<Account> accounts = new List<Account>();
for (List<Account> acts: [SELECT Name FROM Account])
{
        for (Account a : acts) {
                accounts.add(a);
        }
}
```

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Walkthrough: Iterating through a List

STONYPOINT

Action Steps:

Use the Force.com IDE to execute anonymous

1. Write a query to select all Opportunities, their related Products and related Contacts and the Account record associated with the Opportunity
2. Select the results into a loop
3. Within the loop, iterate through each of the lists
4. Print each record to the debug log
5. Save the code in a text editor
6. Check the Debug Log within the Force.com IDE

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Using Variables in SOQL

STONYPOINT

- Square-bracketed queries support binding to Apex variables using a colon (:)

```
String sIndustry = 'Utilities';
List<Account> accounts = [SELECT Name
                          FROM Account
                          WHERE Industry = :sIndustry ];
```

- String queries can be built dynamically

```
String sSOQL = 'SELECT Name FROM Account ';
sSOQL += 'WHERE Industry = \'Utilities\'';
List<Account> accounts = Database.query(sSOQL);
```

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

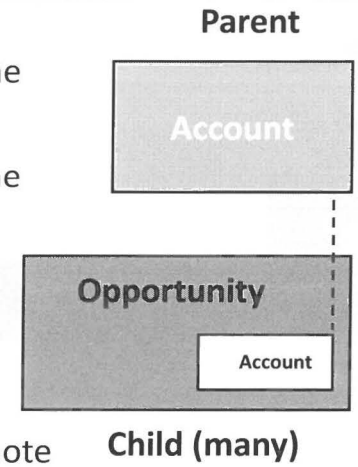
114

Child to Parent in Apex

STONYPOINT

- Two member variables on the child refer to the parent.
  - The foreign key: the unique 18-digit ID of the related record.

```
Opportunity o = new Opportunity();
o.AccountId = '001E000000HfF8E05S';
```



  - The object reference: a reference to the remote sObject object.

```
Opportunity o2 = new Opportunity ();
o2.Account = new Account();
```

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

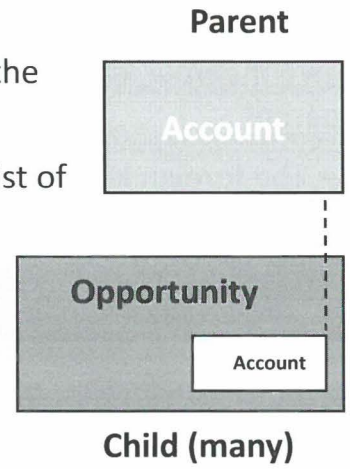
Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Parent to Children in Apex

STONYPOINT

- One member variable on the parent refers to the children:
  - The related list reference: a reference to a List of related sObject objects.

```
Account a = new Account();
a.Opportunities =
        new List<Opportunity>();
```



Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Walkthrough: Creating sObjects in Apex

STONYPOINT

Action Steps:

Use the Force.com IDE to execute anonymous

1. Create a new Account object
2. Create a new Contact object and associate the new Contact with the Account
3. Create a new Opportunity object and associate the new Opportunity with the Account
4. Create a new Opportunity Contact Role object and associate the Contact with the Opportunity
5. Create a list containing 3 Case objects and associate the list of Cases with the Account
6. Save the code in a text editor

Copyright © 2012 Stony Point, Inc. All rights reserved.

Copyright © 2012 Stony Point, Inc. All rights reserved.

DML

STONYPOINT

- Data Manipulation Language
- The only way to modify data in Apex
- Available actions:
  - `Insert`
  - `Update`
  - `Upsert`
    - Actually performs both an Update and Insert
    - Generally should not be used if you are able to separate the new and existing records
  - `Delete`
  - `Undelete`
  - `Merge` (for select objects)

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Invoking DML

STONYPOINT

- DML can be invoked in two ways
  - DML Standalone statements:

```
Account a = new Account();
a.Name = 'ABC Labs';
insert a;
```

  - Database methods:

```
Account a = new Account();
a.Name = 'ABC Labs';
Database.SaveResult sr = Database.insert(a);
```

- Both patterns can be used on either a single `sObject` or lists of sObjects

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

DML Standalone Statements

STONYPOINT

- Single Record Example:

```
Account a = new Account();
a.Name = 'ABC Labs';
insert a;
```

  – If an error is encountered, a run-time exception is thrown

- Multiple Record Example:

```
List<Account> acts = [Select Name from Account];
delete acts;
```

  – If an error is encountered with one record, all records fail.

  – A run-time exception is thrown

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Database.SaveResult

STONYPOINT

- Returned with a `Database.insert()` or `Database.update()` method.
- There are similar `-Result` classes for the other operations
  - Example: `DeleteResult`
- If a single record is submitted, then a single `Database.SaveResult` object is returned
  - A result will always be returned on success or failure
- If a list of records was submitted, then a `List<Database.SaveResult>` list object is returned.
  - Each record in the result list corresponds to a record in the submitted list

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Database Method Calls

STONYPOINT

- Single Record Example:

```
Account a = new Account();
a.Name = 'ABC Labs';
Database.SaveResult sr = Database.insert(a);
```

  - If an error is encountered, no run-time exception is thrown
  - The SaveResult is populated on success or failure

- Multiple Record Example:

```
List<Account> acts = [Select Name from Account];
List<Database.SaveResult> srs = Database.update(acts);
```

  - If an error is encountered with one record, all records fail.
  - A list of SaveResults is populated

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.
122

Database allOrNone

STONYPOINT

- When using the static `Database` methods, there is an optional Boolean parameter called `allOrNone`
  - The value defaults to `true`.
    - Example:

```
Database.insert(accounts);
```

      - `allOrNone = true`
  - Can be explicitly set to `false`
    - Example:

```
Database.insert(accounts,false);
```

    - If `false`, record errors are handled on a case-by-case basis, and the process continues despite having an error in a record.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

SaveResult Methods

STONYPOINT

A `SaveResult` object has three methods available:

- `getId()`
  - Returns the ID of the record that caused the error
- `getErrors()`
  - Returns a List of Database.Error objects
- `isSuccess()`
  - Indicates whether or not the DML operation was successful

```
Account a1 = new Account(name = 'Bank One');
Account a2 = new Account(name = 'Bank Two');
List<Account> accounts = new List<Account>  {a1, a2};
List<Database.SaveResult> srs = Database.insert(accounts,false);
for(Database.SaveResult sr:srs){
        if(!sr.isSuccess()){
                for(Database.Error e : sr.getErrors()) {
                        System.Debug(e.getMessage());
                }
        }
}
```

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

sObject Relationships in Apex

STONYPOINT

- Relationship fields (Lookup & Master Detail) actually expose two interfaces in Apex
  1. The actually ID of the record being referenced
  2. An in-memory reference to the object of the related record

```
Contact c = [Select c.LastName
         from Contact c Where c.Id = '003E000000ECVTC'];
Account a = new Account(name = 'Team Sports');
insert a;
c.Account = a;
update c;
//the contact is not associated with Team Sports
a = new Account(name = 'Golf Team');
insert a;
c.AccountId = a.Id;
update c;
//the contact is now associated with Golf Team
```

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

125

DML and Loops

STONYPOINT

- Force.com has strict Governor Limits on the number of DML statements that can be executed within a single thread
- To avoid hitting Governor Limits when working with lists of records, move the DML statements outside of loops
- Example of improper design:

```
FOR(Account a :[SELECT id FROM Account]) {
  Database.update(a);
}
```

- Example of proper design:

```
List<Account> accounts = new List<Account>();
FOR(Account a :[SELECT id FROM Account]) {
  accounts.add(a);
}
Database.update(accounts);
```

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

DML and Batch

STONYPOINT

- Salesforce provides a technology called Batch Processing that allows for much higher governor limits
- Consider using Batch Processing when the volume of data needing to be processed exceeds the Governor Limit thresholds
- **NOTE**: Batch must be called asynchronously
- In order to take advantage of the higher Batch limits, DML statements must follow this design pattern:

```
for (List<Account> accounts: [SELECT Industry FROM Account])
{
        for (Account a : accounts) {
                a.Industry = 'Construction';
        }
        update(accounts);
}
```

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Walkthrough: DML

STONYPOINT

Goal:
Standardize the naming convention on all Opportunities

Action Steps:
1. Deactivate the Workflow Rule called Set Opportunity Name

Use the Force.com IDE to perform the following:

2. **Hint**: Use the code from the "iterating through a list" section as a starting point
3. Write a query to select all Opportunities
4. Select the results into a loop
5. Within the loop, set the name of the Opportunity to the [Account Name] – [Type] – [Close Month]/[Close Year]
6. At the end of the loop, update the Opportunities list
7. Execute the code and see that the names have been changed
8. Reactivate the Workflow Rule called Set Opportunity Name

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

reactivating workflow - Because, we have a workflow on opp. name & when we try to insert opp.name using DML, it will hit validation rules.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Triggers

STONYPOINT

- Fire automatically when a record is saved
  - 7 events available
    - Before: Update, Insert, Delete
    - After: Update, Insert, Delete, Undelete
- Run as system
- Process in bulk
- Can call classes
- Cannot be called directly
  - The only way to cause code to execute is to invoke a DML event

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Trigger Syntax

STONYPOINT

Trigger declarations use the pattern below:

```
trigger <triggerName> on <ObjectName> (<trigger_events>) {
  <code_block>
}
```

<trigger_events> is a comma-separated list with possible values of:
before insert, before update, before delete, after insert, after update, after delete, after undelete

- Example of a trigger declaration for the Opportunity object

```
trigger setOpportunityName on Opportunity(before insert,
        before update) {
}
```

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

when a trigger is fired - This is the order

1) Validation rules
2) Befor triggers
3) System validations & custom validations
4) Database op's - (ID assigned, record created etc)
5) After insert op's -

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Trigger Context Variables

STONYPOINT

Code running within a Trigger has access to system variables that make it easier for the developer

- `isInsert`, `isDelete`, etc...
  - Allows the developer to interrogate what caused the trigger to fire
- `trigger.new` (upto 200 records)
  - The list of records that caused the trigger to fire
  - Only available on insert, update and undelete triggers
- `trigger.newMap`
  - The map of the records that caused the trigger to fire
  - Only for `before` or `after update` or `after insert` triggers.
- `trigger.old`
  - The list of old values being changed or deleted
  - Only for `update` and `delete` triggers.
- `Trigger.oldMap`
  - The map of old values being changed or deleted
  - Only for `update` and `delete` triggers.
- Check the documentation for additional context variables

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Trigger Considerations

STONYPOINT

- Triggers can cause other triggers to fire
  - The execution of these triggers is part of the same thread and subject to governor limits.
- Upsert events cause `insert` and `update` triggers.
  - This can double execution and cause Governor Limit exceptions
- Merge events fire `delete` triggers for the losing records and then update triggers for the winner.
- All triggers run as System by default.
- All triggers are bulk triggers and must be coded to process up to 200 records per invocation
- The only way to stop a delete is with a trigger
- The ID value is not available in the `before insert` event
- Some built-in system events may ignore triggers you develop
  - Example: Cascade deletes started by deletion of an Account

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Limits Class

STONYPOINT

- If executing code exceeds the allowed Governor Limits, an exception that CANNOT BE TRAPPED is thrown
  - To help you avoid this, Force.com provides a Limits class
- The Limits class provides methods to interrogate the available limits and the limits already consumed
  - Examples:
    - `Limits.getQueryRows()` returns the number of rows already queried
    - `Limits.getLimitQueryRows()` returns the total number of rows allowed to be queried
- The example below demonstrates how to code a query to ensure it does not exceed Governor Limits

```
List<Contact> contacts = [SELECT Name FROM Contact
        LIMIT :(Limits.getLimitQueryRows() - Limits.getQueryRows())];
```

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

addError()

STONYPOINT

- A developer can prevent DML operations from completing by using the `addError()` method
- `addError()` can be added to an instance of an sObject.
  - Can be applied at the sObject level
    - In the standard UI, the error message will appears at the top of the page
  - Can applied at the field level to multiple fields on the same sObject instance
  - In the standard UI, the error message appears next to the field(s) to which it was applied
- Example:

```
FOR(Account a : trigger.new) {
        a.addError('Bad Account');
        a.Name.addError('Invalid Naming Convention');
}
```

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Business Scenario: Naming Conventions

STONYPOINT

When GenWatt deployed Salesforce, they agreed upon a standard naming convention for Opportunity records and trained the sales team on how to name the records. The sales team attempted to follow the convention, but within weeks of deployment the convention broke down.

Earlier you developed a Workflow rule to solve this problem.

After deployment, a shortcoming was identified in the Workflow Rule solution. If the name of the Account record changed, all of the Opportunities did not get updated with the new name of the Account.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Walkthrough: Writing a Trigger

STONYPOINT

Goal:

Develop a trigger on the Account record to change the name of related Opportunity records if the name of the Account changes

Action Steps:

1. Create a Trigger called updateOpportunityFromAccount on the after update event of the Account object
2. Instead of querying the db, use trigger.new as primary list
3. Use the trigger.newMap to find all related Opportunity records
4. Put all the related Opportunity records in a list
5. Update the list

Question: Why isn't it necessary to write any code to actually change the name of the Opportunities?

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

136

Order of Execution

STONYPOINT

1. System validation rules execute.
2. All `before` triggers execute.
3. System validation runs again along with custom validation rules.
4. The record is saved to the database but not committed.
   a. A new record gets a Record ID at this point.
5. All `after` triggers execute.
6. Assignment rules execute (Leads & Cases only).
7. Auto-response rules execute (Leads & Cases only).
8. Workflow rules execute.
   a. Emails, tasks, and Outbound Messages are queued until everything commits at the end
   b. If any Workflow Field Updates occur, the `before` and `after` triggers fire once more
9. Escalation rules execute (Cases only).
10. Rollup summary fields are calculated causing an update on the parent.
    a. If the parent has any automation, all of it will fire at this point as well.
11. All operations are committed to the database.
12. Post-commit logic executes, such as sending email.

* **Important:** If any step along the way fails, everything fails and no changes are saved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Debugging

STONYPOINT

- Allows a developer or administrator to see the details of any transaction that utilizations automation
- Designed to help solve an immediate problem
- Not designed to record and log everything that occurs in the system
  - Unlike other platforms, there is no such thing as a perpetual system log
- Allows debugging for specific Users for a brief period of time
  - You can log up to 20 transactions per User before debugging automatically turns off for that User
- **Path:** Setup -> Monitor -> Logs -> Debug Logs

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Walkthrough: Debug Yourself

STONYPOINT

Action Steps:

1. Navigate to Setup -> Monitor -> Logs -> Debug Logs
2. Create a new Monitored User for your User record
3. Change the name of an Account record and check the Debug Log to see what occurs

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Workflow & Triggers

STONYPOINT

- Using Workflow Field Updates and Triggers on the same object is not recommended
  - Increases the likelihood of exceeding Governor Limits
- Using Workflow Email Alerts, Outbound Messages and Create Task actions has no impact on Governor Limits
- Validation Rules have no impact on Governor Limits

- **Best Practice:** Once Apex Triggers are introduced on an Object, replace any Workflow Field Updates on that Object with Apex Triggers

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Walkthrough: Writing a Trigger

STONYPOINT

Goal:

Develop a trigger on the Opportunity object to enforce a standard naming convention

Action Steps:

1. Deactivate the Workflow Rule called called Set Opportunity Name
2. Create a Trigger called updateOpportunityName on the before update and before insert events of the Opportunity object
3. Instead of querying the db, use trigger.new as primary list
4. Loop through the list
5. Update the name on each record
   - Use the naming convention from the DML walkthrough exercise

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Business Scenario: Apex Triggers

STONYPOINT

GenWatt wants new Invoice records to be created and shared automatically. This cannot be accomplished with Workflow.

The business requirements are that new Invoice records should be created when an Opportunity is set to Closed – Won.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

if / else Statements

STONYPOINT

- The only branching statements available in Apex
  - No CASE statement
- Pattern is `if, else, else if`

```
String color = 'Red';
if(color == 'Green') {
        System.Debug('Go');
} else if (color == 'Yellow') {
        System.Debug('Speed Up!');
} else if (color == 'Red') {
        System.Debug('Stop!');
} else {
        System.Debug('Invalid color');
}
```

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Walkthrough: Writing a Trigger

STONYPOINT

Goal:

Develop a trigger on the Opportunity record to create a new Invoice record when an Opportunity is set to Closed - Won

Action Steps:

1. Create a Trigger called createInvoiceFromOpportunity on the after update and the after insert events of the Opportunity object
2. Instead of querying the db, use trigger.new as primary list
3. Create a list for new Invoice records
4. Check that the Opportunity is Closed & Won and use trigger.OldMap to see if the status of the Opportunity changed
5. If a new Invoice needs to be created, create one and add it to the list of new Invoices
   a. Give it a Due Date of Close Date + 30 days
6. Insert the list into the database

Question: Is the trigger missing any necessary logic?

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Business Scenario: Sharing Records

STONYPOINT

GenWatt is happy with the automatic generation of Invoice records but visibility is not set correctly. Right now, all Users can see all Invoices.

GenWatt management requested that only certain people should be able to see invoices. The invoice should be visible to the sales person who owned the Opportunity and all of customer service. Customer service should not be able to see the Opportunity record associated with the invoice, just the invoice.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Exercise: Organization Wide Defaults

STONYPOINT

Goal:

Using Organization Wide Defaults, lock down the Invoice record so that Users only see the records that they own.

Action Steps:

1. Navigate to Setup -> Administer -> Security Controls -> Sharing Settings
2. Edit the Organization Wide Defaults and set the Invoice object to Private

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Apex Sharing

STONYPOINT

- Apex sharing is the automation of manual sharing.
- Used when other sharing automation techniques cannot fully address the business need
- Custom Apex Sharing Reasons
  - Can only be created on custom objects
  - Allows the User to understand why a particular user has access to a particular record

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Exercise: Apex Sharing Reason

STONYPOINT

Goal:

Create an Apex Sharing Reason for the Invoice object.

Action Steps:

1. Navigate to Setup -> Build -> Create -> Objects
2. Click on the Invoice Object and navigate to the Apex Sharing Reason section.
3. Create a new Apex Sharing Reason with the following attributes:
   - Reason Label: Opportunity Owner
   - Reason Name: Opportunity_Owner
4. Save the Apex Sharing Reason
5. Create another Apex Sharing Reason with the following attributes:
   - Reason Label: Customer Service
   - Reason Name: Customer_Service

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Sharing Tables

STONYPOINT

- Whenever the Organization Wide Default setting for an Object is anything less than Public Read/ Write, Salesforce automatically creates a __Share table for that object
  - Example: Invoice__Share
- The table has 4 important fields:
  - ParentId: The Id of the record being shared
  - UserOrGroupId: The Id of the User record or the Group record that the ParentId is being shared with
    - The GroupId can also refer to a Role because Salesforce automatically creates a hidden Group for each Role
  - AccessLevel: The level of access being granted to the UserOrGroupId
    - Possible Values: Read, Edit, Owner
  - RowCause: The reason the record is being shared
    - There are quite a few standard reasons
    - For custom objects, Apex Sharing Reasons are also available

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Walkthrough: Writing a Trigger

STONYPOINT

Goal:

Develop a trigger on the Invoice object to ensure the appropriate users have proper visibility to invoice records

Action Steps:

1. Create a Trigger called setInvoiceSharing on the after update and the after insert events of the Invoice object
2. Create a list for new Invoice__Share records
3. Create a Set to store the Opportunity Ids
4. Create a Map to store the Opportunity Ids and the Opportunity Owner Id
   a. Query the database to populate the Map
5. Query the database to find the SVP, Customer Service role
6. Iterate through the new Invoice records
   a. Create new share record for Opportunity Owner
   b. Create new share record for SVP
7. Insert the new Share records into the database

Question: Is the trigger missing any necessary logic?

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.
150

Classes

STONYPOINT

- Apex supports most object oriented tenants through classes
- Classes support inheritance
- Classes support interfaces
- Classes can have properties and methods
- Classes can respect security settings
- Class methods can be overloaded
- Classes support different levels of external access
  - Private, Public, Global
- All Public classes within in a Salesforce namespace are automatically visible to all other classes within the namespace
  - No need to explicitly include file paths or headers in your code
- Classes are typically used as:
  - Unit Tests
  - Visualforce controllers
  - SOAP Webservices
  - REST Webservices
  - Email Services
  - Scheduled Batches

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Class Access Modifiers

STONYPOINT

```
private | public | global
[virtual | abstract | with sharing | without sharing | (none)]
class ClassName [implements InterfaceNameList | (none)] [extends
ClassOrInterfaceName |(none)]{
  // The body of the class
}
```

- `global`: This class may accessible outside of your namespace.
  - Methods must be exposed either as `global` or as `webservice`
  - `global` methods not `webservice` are only accessible when installed as part of a managed package
- `public`: This class is visible across your application or namespace.
- `private`: This class is an inner class and is only accessible to the outer class, or is a test class.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Class Inheritance

STONYPOINT

```
private | public | global
[virtual | abstract | with sharing | without sharing | (none)]
class ClassName [implements InterfaceNameList | (none)] [extends
ClassName |(none)]{
  // The body of the class
}
```

- The `virtual` modifier declares that this class allows extensions
  - Classes not using the `virtual` modifier cannot be extended
- The `extends` keyword is used to extend a parent class and create a subclass.
  - The extending subclass can:
    - Extend the virtual class with new methods
    - Override existing methods in the virtual class
    - Access the original methods from parent class using the `super` keyword

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Class Sharing

STONYPOINT

```
private | public | global
[virtual | abstract | with sharing | without sharing | (none)]
class ClassName [implements InterfaceNameList | (none)] [extends
ClassName |(none)]{
  // The body of the class
}
```

- Apex generally runs with System permissions
  - Triggers always run as System
- Classes have the ability to explicitly declare whether or not to run as System or as the invoking user
  - It is not possible to run code as or impersonate another user
  - `with sharing`: will respect all sharing and security settings of the invoking User
  - `without sharing`: will ignore all sharing and security settings and will run as system
  - No declaration: will inherit the context of whatever is calling the class
    - Example: When a trigger calls a class with no sharing declaration, the class will run as System since the trigger is running as System

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.
154

Member Variables

STONYPOINT

- Member variables must be declared
- Member variables are strongly typed
  - `Var x;` – not valid
  - `String x;` - valid
- Support access modifiers
  - Declarations without an access modifier default to `private`
  - `private:` Accessible only within the class where it is defined
  - `protected`: Available to any inner classes or subclasses.
    - Can only be used by instance methods and member attributes
  - `public`: Can be called from anywhere in the namespace.
  - `global`: May be accessible by Apex outside of the namespace
- Values can be set during declaration

```
String s = 'Stony Point';
Account a = new Account();
```

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Class Properties

STONYPOINT

- Apex supports two design patterns for properties
  - Getters & Setters

```
String sFoo;

public String getFoo() {
        return sFoo;
}

public void setFoo(String s) {
        sFoo = s;
}
```

  - Get; Set;

```
Public String foo {get;set;}
```

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Methods

STONYPOINT

- Can accept input parameters
- Support the same access modifiers as member variables
- Can return a value
  - Support void as a return type
- Can be static methods

```
public static Integer addIntegers(Integer x, Integer y) {
    return x + y;
}
```

- Can be instance methods

```
Public Integer addInts(Integer x, Integer y) {
    return x + y;
}
```

- Can be overloaded
  - Signature must be unique

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Walkthrough: Teenager Text Simulator

STONYPOINT

Goal:

Develop a class that will simulate having a text message conversation with a teenager

Action Steps:

1. Create a class called TeenagerText
2. Create a map of possible teenage text responses
   a. Load the map in the constructor of the class
3. Create a method that accepts an incoming string and returns a string from the map of possible responses
4. Test the class using execute anonymous

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Unit Tests

STONYPOINT

- 75% of all code must be covered by unit tests
- Every class doesn't need to be covered
- Every trigger must have at least 1% code coverage
- To test a trigger, write a test method that executes a DML statement

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Walkthrough: Writing a Unit Test

STONYPOINT

Action Steps:

Develop a test class to test the triggers previously written

1. Create a test class
2. Write a test method to test the name change
3. Create an Account record
4. Create an Opportunity record and set it's status to Closed-Won
5. Assert that the Opportunity name field is properly set
6. Query the database to ensure that a new Invoice record was created
7. Assert that an Invoice record was created

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Visualforce

STONYPOINT

- A proprietary web development language created by salesforce.com
  - Syntax for language is XHTML
    - Ultimately renders the page in HTML
- Not another name for Apex, a different technology
  - Typically used in conjunction with Apex, but does not require Apex
  - Fits into the View level of MVC
- Allows developers to build custom web pages to run on the Force.com platform
  - Not intended for pure custom website creation
  - Strength is that it allows developers to easily create a completely custom page that has the exact look and feel of a standard Salesforce page
- A web design or web development skill set is required to create
  - Currently no good native or 3rd party tools available
    - This is changing with SiteForce and site.com
  - Most developers will use the Force.com IDE

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

When to use Visualforce

STONYPOINT

- Almost always used in conjunction with a custom Apex controller
- To display records from multiple objects
- To edit multiple records at the same time
- To create multiple records at the same time
- Custom analytic capabilities
- When the declarative interface can't do what you want

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.
162

Tag Syntax

STONYPOINT

- Visualforce tags (also referred to as components) all begin with the `apex:` prefix.
  - Example: `<apex:dataTable>`
- All tags must have both an opening and closing tag (XHTML format)
  - Valid: `<apex:dataTable></apex:dataTable>`
  - Valid: `<apex:dataTable/>`
  - Invalid: `<p>`
  - Valid: `<p></p> or <p/>`
- All tag attributes must be encapsulated in “”
  - Valid: `<table width="400">`
  - Invalid: `<table width=400>`
- All pages must be enclosed by a set of `<apex:page> </apex:page>` tags.
- Some tags support embedded HTML
  - `<apex:outputPanel><b>Hello World</b></apex:outputPanel>`

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Visualforce Key Tags

STONYPOINT

The best place to get detailed information on Visualforce is from the Visualforce Developer's Guide on developer.force.com.

Below is a list of the most commonly used tags. You will learn about tags in more detail in the exercises that follow.

- `<apex:page>`
- `<apex:sectionHeader>`
- `<apex:pageBlock>`
- `<apex:pageBlockSection>`
- `<apex:pageBlockButtons>`
- `<apex:pageBlockTable>`
- `<apex:column>`
- `<apex:facet>`
- `<apex:form>`
- `<apex:inputField>`
- `<apex:outputField>`
- `<apex:commandButton>`
- `<apex:commandLink>`
- `<apex:param>`

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Exercise: Visualforce Developer's Guide

STONYPOINT

Goal:

Create a bookmark to the Visualforce Developer's Guide and explore the Standard Component Reference section

Action Steps:

1. In a new browser window, navigate to http://developer.force.com
2. In the Search bar, type in Visualforce Developer's Guide.
3. The first search result should be the guide.
4. Click on the link and create a bookmark to the page that comes up
5. In the navigation bar on the left, find the section that says Standard Component Reference

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Visualforce Controllers

STONYPOINT

There are three types of Visualforce controllers

- Standard Controllers
  - Automatically available for all sObjects, both Standard and Custom
  - Standard functionality
    - Retrieve a record and its related lists
    - Create, update, delete, clone
- Controller Extensions
  - Apex class that works in conjunction with Standard Controller
  - Must use for overrides and standard UI integration
- Custom Controllers
  - Apex class with completely custom functionality

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Tag Bindings

STONYPOINT

- Bindings relate Visualforce tags with
  - The page controller
  - Other tags on the Visualforce page
- There are three types of bindings on Visualforce pages:
  - Data bindings
    - Pull in data from the data made available by the controller
    - Can be bound to sObject fields or properties on the controller
  - Action bindings
    - Call functions in the controller
  - Component bindings
    - Reference other components on the page
    - Used when writing custom Javascript on the page

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Binding Syntax

STONYPOINT

- Data Binding
  - `<apex:inputField value="{!Position__c.Name}" />`
  - `<apex:outputText value="{!$User.Name}" />`
  - `<apex:dataTable value="{!Account.Contacts}">`
- Action Binding
  - `<apex:commandButton action="{!save}" />`
  - `<apex:commandLink action="{!view}" />`
  - `<apex:actionPoller action="{!refresh}" />`

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Component Binding Example

STONYPOINT

```
<apex:page id="thePage">
  <script>
    function changeFont(input, textid) {
      if(input.checked) {
        document.getElementById(textid).style.fontWeight = "bold";
      }
      else {
        document.getElementById(textid).style.fontWeight = "normal";
      }
    }
  </script>

  <!-- This outputPanel calls the function, passing in the checkbox and the ID of the target component. -->
  <apex:outputPanel layout="block">
    <label for="checkbox">Click this box to change text font:</label>
    <input id="checkbox" type="checkbox"  onclick="changeFont(this,'{!$Component.thePanel}');"/>
  </apex:outputPanel>
  <!-- This outputPanel is the target, and contains text that will be changed. -->
  <apex:outputPanel id="thePanel" layout="block">
    Change my font weight!
  </apex:outputPanel>
</apex:page>
```

*Source: Visualforce Developer's Guide: Using JavaScript to Reference Components

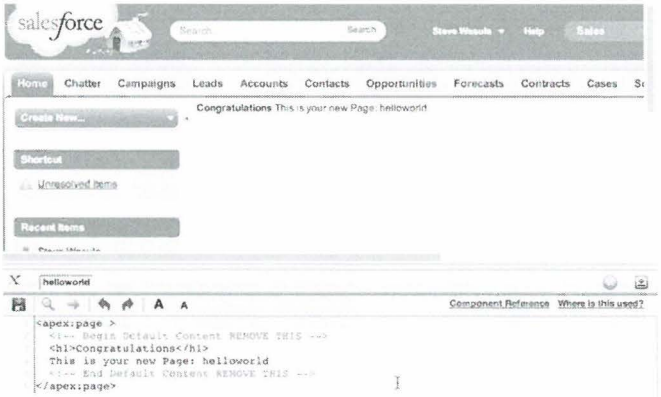
Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Development Mode

STONYPOINT

- Enables System Administrators and Developers to:
  - Create new Visualforce pages simply be entering a URL such as: https://na9.salesforce.com/apex/helloworld
  - Edit Visualforce pages using a split code window as shown below:



- In order to create or edit Visualforce pages a User needs the "Customize Application" Profile permission

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Exercise: Enable Development Mode

STONYPOINT

Goal:

Enable Development Mode and other User Settings

Action Steps:

1. Navigate to [Your Name] -> My Settings -> Personal -> Advanced User Details
2. Click the ***Edit*** button to edit your User record
3. Make sure the following settings are enabled:
   a. Development Mode
   b. Show View State in Development Mode
4. Save your record
5. You will see the impact of Development Mode during the Visualforce exercises.

NOTE: Once you start developing custom controllers and complex Visualforce pages, Stony Point recommends that you disable Development Mode and perform all coding in the Force.com IDE

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

171

Walkthrough: Visualforce

STONYPOINT

Goal:

Create your first Visualforce page

Action Steps:

1. Create a new page by entering a new URL: https://na[x].salesforce.com/apex/teenager
2. To make the page appear like other detail pages, add a sectionHeader tag.
   a. Set the title attribute to "Teenager Texting"
   b. Save the page to see the changes
3. Add a pageBlock tag and save the page to see the changes
4. Within the pageBlock tag, add a pageblockSection tag and save the page
5. Within the pageBlockSection tag, add a greeting to yourself on the page by adding the following tag:

   ```
   <apex:outputText value="Hello {!$User.FirstName}" />
   ```

6. Save the page

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Inserting Visualforce Pages into Salesforce

STONYPOINT

- Internally within Salesforce
  - Visualforce Tabs
  - Button Overrides
  - Custom Buttons
  - Stand Alone Pages (URL Link)
  - Custom Help Pages
  - Dashboard Components
  - Embedded into a page layout
  - Home Page Layout
- Partner Portal / Customer Portal (sunset)
- Salesforce Communities
- Force.com Sites

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Walkthrough: Visualforce Tab

STONYPOINT

Goal:

Create your first Visualforce custom tab

Action Steps:

1. Navigate to Setup -> Create -> Tabs
2. Create a new Visualforce tab with the following attributes:
   a. Visualforce Page: teenager
   b. Tab Label: Teenager
   c. Tab Name: Teenager
   d. Tab Style: Cell Phone
3. Set the Tab Visibility to Default On for all profiles
4. Add the tab to all applications
5. Save the tab

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Button Overrides

STONYPOINT

- Used to insert Visualforce pages into Salesforce
- A button override globally modifies navigation
  - Example: If the Edit button is overridden, every navigation to the edit will go to the override page
- Button overrides replace both buttons and links
- Once a button is overridden, standard page layouts cannot be used for the overridden action
- There are eight "buttons" that can be overridden
  - Tab
  - List
  - View
  - Edit
  - Delete
  - Clone
  - New
  - Accept

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Walkthrough: Override a Button

STONYPOINT

Goal:

Create a new Account Detail page and override the standard Account view behavior

Action Steps:

1. Create a new page by entering a new URL: https://na[x].salesforce.com/apex/accountdetail
2. Update the Visualforce Page code to look like the code below:

```
<apex:page standardController="Account">
  Hi {!$User.FirstName}, this is the new, slightly less functional Account Detail page
  <apex:detail subject="{!Account.Id}"/>
</apex:page>
```

3. Override the View button on the Account by navigating to Setup -> App Setup -> Customize -> Accounts -> Buttons, Links and Actions
4. Click on the ***Edit*** link next to the View button
5. Override the button with the accountdetail Visualforce Page
6. Navigate to any Account record and see your new page in action

Questions:

1. How is this slightly less functional?
2. Why didn't the Teenager Visualforce page show up in the drop down list?

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Business Scenario: Improve a Custom Tab

STONYPOINT

Some of your end users have complained about the behavior of the tabs you developed for Invoices. They have asked that the tab immediately navigate to the last view they used. This behavior is not possible with a custom object tab, but it is possible with Visualforce.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Walkthrough: Override a Tab

STONYPOINT

Goal:

Create a new Invoices tab page and override the custom object tab currently in place

Action Steps:

1. Create a new page by entering a new URL: https://na[x].salesforce.com/apex/invoicestab
2. Modify the page to look like the code below:

```
<apex:page tabstyle="Invoice__c">
 <apex:enhancedList height="800" width="600" rowsPerPage="50"
type="Invoice__c"/>
</apex:page>
```

3. Override the current custom object tab by navigating to Setup -> App Setup -> Create -> Objects -> Invoice
4. On the Invoice object definition, navigate to the Buttons, Links and Actions related list and click the ***Edit*** link next to the Invoices Tab
5. Override the button with the invoicestab Visualforce Page
6. Save the changes
7. Navigate to the Invoices tab and your Visualforce Page will display

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Business Scenario: PDF's

STONYPOINT

Some of your customers and users have requested that you provide them with the ability to easily create invoices in PDF format. The users would like the option of seeing the invoice record on screen in normal format, but would also like to be able to generate a pdf of the invoice to save to their desktop or print.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Walkthrough: PDF's

STONYPOINT

Goal:

Create a new Visualforce page that will generate an Invoice in PDF format

Action Steps:

1. Create a new page by entering a new URL:
   https://na[x].salesforce.com/apex/invoicepdf
2. Modify the page to look like the code below:

```
<apex:page standardController="Invoice__c" renderas="pdf">
    <apex:detail subject="{!Invoice__c.Id}"/>
</apex:page>
```

3. Save the page
4. Navigate to Setup -> App Setup -> Create -> Objects -> Invoice
5. On the Invoice object definition, navigate to the Buttons, Links and Actions related list and create a new button with the following attributes:
   a. Label: PDF
   b. Name: PDF
   c. Display Type: Detail Page Button
   d. Behavior: Display in new Window
   e. Content Source: Visualforce Page
   f. Content: invoicepdf
6. Save the button

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Walkthrough: PDF's

STONYPOINT

Action Steps (continued...):

7. Navigate to Setup -> App Setup -> Create -> Objects -> Invoice
8. On the Invoice object definition, navigate to the Page Layouts section and edit the Invoice Layout page layout
9. Add the PDF button to the Custom Buttons section on the page layout
10. Save the layout
11. To test the button, navigate to the Invoices tab and select any Invoice record
12. Click on the PDF button and a new window should open with the Invoice Detail displayed in PDF format

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Design Notes: PDF's

STONYPOINT

- PDF's do not automatically have style sheets
- Apply your own style sheets
- Embed the style sheets into the page instead of linking to them
  - Linked style sheets have inconsistent behavior

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Business Scenario: Multiple New Invoices

STONYPOINT

The trigger you developed earlier to automatically create invoice records when an opportunity is working great. However, about 10% of the time, customers negotiate payment plans and need to be invoiced multiple times for the same opportunity. Typically, the customers are invoiced monthly or quarterly and the amount of each invoice may vary.

The finance users have asked for the ability to easily create up to 12 new invoice records on a single screen.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Walkthrough: Adding Complex Functionality

STONYPOINT

Goal:

Develop a Visualforce page, controller extension, and custom list button for Invoices that will allow a User to create multiple Invoice records at the same time. Add the list button to the Opportunity page layouts

Action Steps:

1. Create a class called InvoicesController
2. Create the shell for the constructor to support being an extension
   a. Hint: Use the StandardSetController
3. Create a new Visualforce page called new_invoices that is bound to the Invoice__c standard set controller and the InvoicesController extension
4. Create a new custom list button on the Invoice object and bind it to the new_invoices Visualforce page
5. Add the custom list button to the Invoices related list on all Opportunity page layouts

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Walkthrough: List Controller Extension

STONYPOINT

Action Steps:

1. Build a property that creates an empty list of 12 Invoice records to bind the page to
2. Build a save method that saves any new records with an amount greater than $0
3. Build a cancel method

The following steps apply to the new_invoices Visualforce page

4. Add an apex:form tag and an apex:pageblock tag
5. Add an apex:pageblockbuttons tag with save and cancel command buttons
6. Add an apex:pageblocktable to the page and bind it to the invoices property in the controller extension
7. Add apex:columns to display the invoice name, Account, and Opportunity
8. Add an apex:inputText field to capture the Amount field
9. Test the page by navigating to any Opportunity record and launching the new_invoices Visualforce page from the Invoices related list
   a. If coded correctly, only Invoices with amounts > $0 will be created

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Business Scenario: Editing Multiple Invoices

STONYPOINT

The Visualforce page you just created is working great and now the finance department has asked for an additional enhancement. It seems that sometimes they create multiple invoices with incorrect amounts.

The users would like to be able to edit all existing invoices just as easily as they can add new invoices.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Walkthrough: Editing Multiple Records

STONYPOINT

Action Steps:

1. Update the SOQL code to retrieve the existing Invoice records when retrieving the AccountId
2. Build a property that exposes the list of existing invoices to bind the page to
3. Update the save method to check whether it is getting new or existing records
4. Create a new Visualforce page called existing_invoices
   a. Since the page does almost exactly the same thing at the new_invoices page, copy the code
4. Ensure you have columns to display the invoice name and the balance
5. Ensure that the Account, Opportunity, Amount and Amount Paid fields are editable
6. Create a custom list button named Edit All Invoices on the Invoice object and add it to all Opportunity page layouts
7. Test the page by navigating to any Opportunity record that has Invoice records and launching the existing_invoices Visualforce page from the Invoices related list

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Design Discussion

STONYPOINT

- What design challenges would we encounter if we tried to allow a User to edit existing records and create new records on a single page?
- What happens if one invoice record fails to save for some reason?
  - Can this behavior be changed?
- How are error messages displayed in Visualforce?

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Sending Emails with Apex

STONYPOINT

- It is possible to send emails programmatically using Apex
  - **Not recommended**
- External emails sent using Apex count against your daily mass email limit of 500 emails per day
  - That means you can send a maximum of 500 emails to non-Salesforce users
  - When you exceed your limit, the emails just don't get sent
- Email templates don't work as well as they should within Apex
  - The most reliable method is to have the code develop the text of the email
  - This requires a developer to change code for minor changes to the text of the message

BEST Practice: Use Workflow Email Alerts to send email

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Emails with Apex & Workflow

STONYPOINT

- Workflow is the best way to send email but it has limitations as well
  - Tied to a single record
  - Requires a DML event to fire
  - Email templates can only access the fields on a single record
  - Doesn't log that it sent an email

BEST Practice: Use Apex to trigger Workflow Email Alerts to send email

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Business Scenario: Email Automation

STONYPOINT

The GenWatt customer service team has requested an easy way to email a customer an Invoice. If at all possible, they would like to click on a button and have the email sent automatically to the billing contact listed on the Invoice record.

GenWatt is struggling to figure out how to send the email. They don't want to use Apex because they expect the number of requests to exceed 500 per day. They want to use Workflow Email Alerts, but no DML event is taking place and they don't want to modify the Invoice record and change the timestamps just to send an email.

What would you recommend to GenWatt?

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Design Notes: Custom Buttons

STONYPOINT

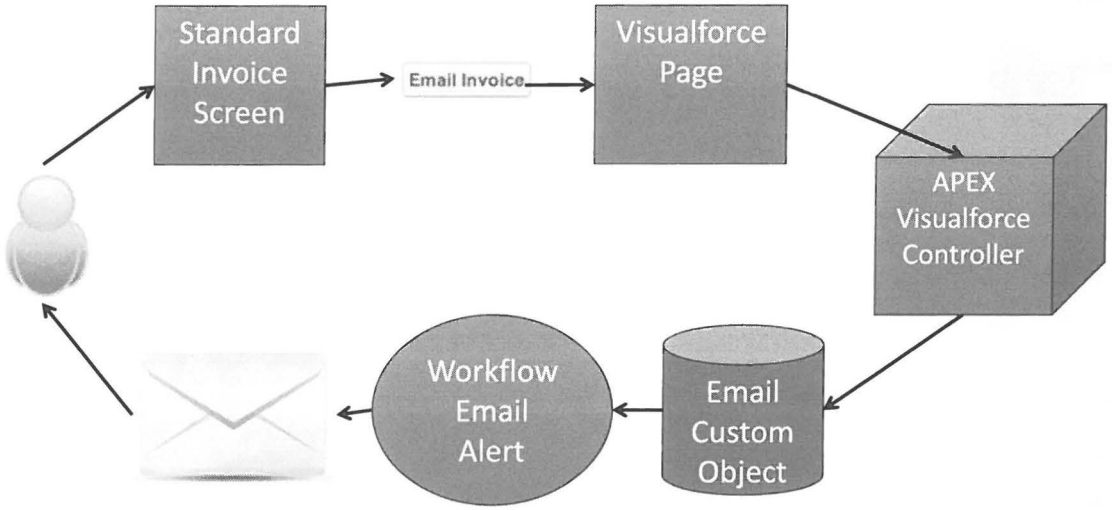
- Custom buttons on standard page layouts cannot call Apex code directly
  - Only Visualforce pages can call Apex code
- Solution: Visualforce transition pages
- Steps in building a transition page
  1. Build a controller extension with an action
  2. Build a Visualforce page that calls the action from the apex:page tag which causes the action to fire immediately
  3. When the action completes, redirect the page back to the original page
- To the end user, it appears as though the button performed the action and they stayed on the same page

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.
192

Custom Button Design

STONYPOINT



Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Exercise: Invoice Custom Fields

STONYPOINT

Goal:

GenWatt has decided it will send the Invoice email to the billing contact associated with the Invoice. Create a Billing Contact field

Action Steps:

1. Navigate to Setup -> Build -> Create -> Objects
2. Click on the Invoice Object and navigate to the Custom Fields & Relationships section. Create the following new fields:

Type	Label	Name	Details
Lookup	Billing Contact	Billing_Contact	Related To: Contact Clear the value of this field. Make Visible to all Profiles Do not add related list to page layouts

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.
194

Exercise: Email Custom Object

STONYPOINT

Goal:

Create a new custom object to trigger email alerts.

Action Steps:

1. Navigate Setup -> Build -> Create -> Objects
2. Create a new custom object by clicking on the New Custom Object button
3. Set the following options:
   a. Label: Email
   b. Plural Label: Emails
   c. Object Name: Email
   d. Record Name: Email #
   e. Data Type: Auto Number
   f. Display Format: EMAIL-{00000000}
   g. Starting Number: 1
   h. Allow Reports
   i. Allow Activities
   j. Deployed
4. Save the object by clicking the Save button.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Exercise: Email Custom Tab

STONYPOINT

Goal:

Create a Custom Tab for the Email object

Action Steps:

1. Navigate to Setup -> Build -> Create -> Tabs
2. Create a new Custom Object Tab with the following settings:
   a. Object: Email
   b. Tab Style: Your Choice of icon and colors
3. Set the Tab visibility to:
   a. System Administrator: Default On
   b. All other Profiles: Default Off
4. Add the Tab to all Apps

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Exercise: Email Custom Fields

STONYPOINT

Action Steps:

1. Navigate to Setup -> Build -> Create -> Objects
2. Click on the Email Object and navigate to the Custom Fields & Relationships section. Create the following new fields:

Type	Label	Name	Details
Lookup	Opportunity	Opportunity	Related To: Opportunity Clear the value of this field. Make Visible to all Profiles Do not add related list to page layouts
Lookup	Account	Account	Related To: Account Clear the value of this field. Make Visible to all Profiles Do not add related list to page layouts
Lookup	Contact	Contact	Related To: Contact Clear the value of this field. Make Visible to all Profiles Do not add related list to page layouts

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Exercise: Email Custom Fields

STONYPOINT

Type	Label	Name	Details
Lookup	Invoice	Invoice	Related To: Invoice Clear the value of this field. Make Visible to all Profiles Do not add related list to page layouts
Formula	Invoice Amount	Invoice_Amount	Return Type: Currency, 2 Decimal Places Formula: Invoice__r.Amount__c Make Visible to all Profiles Add to Email Page Layout

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Exercise: Creating an Email Template

STONYPOINT

Goal:

Create an email template to send an Invoice to a customer

Action Steps:

1. Navigate to Setup -> Administer -> Communication Templates -> Email Templates
2. Create a new Email Template of type Text
3. Choose the following options
   a. Folder: Unfiled Public Email Templates
   b. Available for Use: Checked
   c. Email Template Name: Invoice
   d. Template Unique Name: Invoice
   e. Subject: Invoice # {!Email__c.Invoice__c}
   f. Email Body:
      Invoice #: {!Email__c.Invoice__c}
      Invoice Amount: {!Email__c.Invoice_Amount__c}
4. Save the Template

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Exercise: Workflow Rule

STONYPOINT

Goal:

Create a new Workflow Rule that will use an Email Alert to send an Invoice via email to a customer

Action Steps:

1. Navigate to Setup -> Build -> Create -> Workflow & Approvals -> Workflow Rules
   a. Create a new Workflow Rule
   b. Select the Email object as the object type
   c. Set the following options
      i. Rule Name: Send Invoice
      ii. Evaluate the rule when a record is: created
      iii. Rule Criteria: formula evaluates to true
      iv. Enter the following formula into the formula field
         1=1
   d. Click Save & Next

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Exercise: Workflow Rule

STONYPOINT

Action Steps:

2. Add an immediate Workflow Action by clicking the Add Workflow Action button in the Immediate Workflow Actions section
   a. Add a New Email Alert and set the following options:
      i. Description: Send Invoice
      ii. Unique Name: Send_Invoice
      iii. Email Template: Invoice
      iv. Recipient Type: User
      v. Select yourself
      vi. Recipient Type: Related Contact
      vii. Select Contact
      viii. Save the Email Alert
3. Click the Done button to finish the rule
4. Activate the Rule by clicking the Activate button
5. Test the rule by creating a new Email record making sure to select an Invoice record.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Walkthrough: Visualforce Transition Page

STONYPOINT

Goal:

Create an Apex controller and Visualforce page to create an email record

Action Steps:

1. Create a new Apex class named InvoiceController
2. Add a method to create an email record
3. Create a new Visualforce page called emailInvoice
4. Bind the page to the Invoice standard controller and the InvoiceController extension.
5. Call the EmailInvoice action from the apex:page tag
6. Build a custom detail button on the Invoice object named Email Invoice and add it to the Invoice page layout
7. Test the button by navigating to any Invoice record, making yourself the Billing Contact and clicking on the button
   a. Turn on debugging to see what's happening behind the scenes
   b. You should receive an email and a new Email record should have been created in the Email object

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Setup Audit Log

STONYPOINT

- Every change made to the Salesforce Metadata is automatically recorded in the Setup Audit Log
- Records configuration changes made through
  - The user interface
  - Package installations
  - Change Sets
  - Metadata API
- Salesforce records:
  - Who made the change
  - The change that was made
  - When it was made
- Also records when a System Administrator logins in as another User
- Displays the last 20 changes online
- Allows download of a list of changes made over the 6 months
- Path: Setup -> Administer -> Security Controls -> Setup Audit Trail

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Batch Apex

STONYPOINT

- Used to build complex, long-running processes on the Force.com platform that would exceed normal Governor Limits
  - Allows processing of up to 50 million records in a single batch
- Also used for Apex managed sharing recalculations
- Batch Apex is exposed as an interface
- Batch jobs can be programmatically invoked at runtime using Apex.
  - Are often scheduled using the Apex scheduler
  - To be scheduled, a Batch class must also implement the Scheduleable Inteface
- There are a maximum of five queued or active batch jobs at one time.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Technical Scenario

STONYPOINT

For some reason, some Opportunity names are not getting updated to the proper naming convention. You suspect that every now and then, given a specific set of circumstances that you cannot recreate, Governor Limits are being hit when the triggers are being run. Rather than spend days or weeks to diagnose the true cause, you want to use brute force to solve the problem.

You want to run a job every night to update all Opportunity names and you don't want to have to worry about Governor Limits.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Walkthrough: Scheduling Batch Apex

STONYPOINT

Goal:

Create batch Apex class, create a Schedulable Apex class that calls the batch and schedule the batch to run every night

Action Steps:

1. Create a new Apex class named UpdateOpportunitiesBatch that implements the Database.batchable interface
2. Implement the essential batch methods
3. Select all Opportunity records and update them in the batch
4. Create a new Apex class named UpdateAllOpportunities that implements the Schedulable interface
5. Call the UpdateOpportunitiesBatch class from within the execute method
6. Using the Scheduled Jobs functionality in Salesforce, schedule the UpdateAllOpportunities class to run nightly
   a. Hint: Navigate to Setup -> Develop -> Apex Classes and click on the Schedule Apex button

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Dynamic Pick Lists

STONYPOINT

- Native pick lists in Salesforce have all values predefined and stored in the metadata
- Controlling and dependent pick lists work the same way
- Visualforce allows us to create a pick list on the fly utilizing the <apex:selectList /> component
  - The component is then bound to an Apex property that returns a `List<SelectOption>`
- Typical Usage:
  - Display records as a pick list instead of a Lookup field
  - Display different values based on a selected value

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Business Scenario: Dynamic Pick Lists

STONYPOINT

The GenWatt finance team wants to make it harder to create invoices with invalid data. GenWatt wants to ensure that Invoices can only be created from the related list on Account record and that only Contacts from that Account can be selected as the Billing Contact.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Walkthrough: Dynamic Pick Lists

STONYPOINT

Goal:

Create an Apex controller and Visualforce page to override the standard new invoice page

Action Steps:

1. Create a new class called NewInvoice
2. Create a new Visualforce page called NewInvoice and bind the page to the Invoice standard controller and the NewInvoice extension
3. Modify the constructor to:
   a. Get the AccountId from the input parameters
   b. Find the related Contacts
4. Add the method to the InvoiceController class to create a new Invoice
5. Add a cancel method
6. Add properties to expose the Account and list of related Contacts
7. Override the New button for the Account object
8. Add the fields to the Visualforce page
   a. Change the Billing Contact field to be a SelectList and bind to the Contacts list

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Visualforce Components

STONYPOINT

- Developers can create custom Visualforce Components to be re-used within Visualforce pages
- Typical usage
  - Headers, Footers, Menus
  - Custom user interface elements
    - Radio buttons, multi-select pick lists, calendar controls
- Considerations
  - Can only be utilized in a Visualforce page
  - Can be bound to a custom controller
  - Can have custom attributes
- Are referenced utilizing <c:[componentname] />

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Visualforce Component Tags

STONYPOINT

- <apex:component>
  - Must be used as the first and last tag
  - Key attributes
    - Controller
    - AllowDML – must be enabled if the component will update the database
- <apex:attribute>
  - Allows the Visualforce page to pass parameters to the component
  - Key attributes
    - Type – the data type being passed
    - Required
    - AssignTo – a property in the custom controller to bind the input to
- <apex:componentbody>
  - Allows the developer of the Visualforce page to embed HTML within the open and close tag of the component
  - Without this tag, no HTML can be passed to the component

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Business Scenario: Components

STONYPOINT

The GenWatt legal department has become concerned that employees are not being properly notified when they are viewing protected, proprietary or confidential information within Salesforce.

They have mandated that Accounts, Contacts, Opportunities and Invoices all have disclaimers appear whenever someone views the record.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Walkthrough: Components

STONYPOINT

Goal:

Create a Visualforce Component and embed it on Visualforce pages for display

Action Steps:

1. Create a new Component called Disclaimer
2. Add an attribute for Level and for RecordName
3. Add the following disclaimer "The data you are viewing, concerning the {!RecordName} record, is the sole property of GenWatt, Inc. and should not be disclosed to any 3rd party under any circumstances."
4. Add the component to the AccountDetail page and set the Level to 2
5. Add the component to the NewInvoice page and set the Level to 1
6. Create a Visualforce pages for the Contact and Opportunity pages
   a. Add the component to each page and set the Level to 3
7. Embed the Visualforce pages in the Contact and Opportunity page layouts

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Visualforce in Dashboards

STONYPOINT

- Visualforce pages can be included in Dashboards to allow robust functionality not normally supported
- Considerations
  - To be included in a dashboard the Visualforce page can:
    - Have no controller
    - Have a custom controller
    - Be bound to a StandardSetController
  - A Visualforce page that uses a StandardController cannot be included in a dashboard
  - There are width limitations in a dashboard of about 400 pixels per dashboard column

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.
214

Charts in Visualforce

STONYPOINT

- Visualforce has a variety of components available to duplicate the same charts and graphs that can be generated in reports and dashboards
- Data for charts can come directly from a SOQL query, a custom dataset or from Javascript
- Key components
  - <apex:chart />
    - Main component which defines the data and size of chart
  - <apex:axis />
    - Sets the axis attributes for a chart
  - <apex:barSeries />, <apex:lineSeries />, <apex:pieSeries />, <apex:gaugeSeries />, <apex:radarSeries />
    - Defines the type of chart being rendered and its appearance
  - <apex:legend />
    - Defines the legend attributes for a chart

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.
215

Wrapper Classes

STONYPOINT

- A concept, not a construct
- An Apex class typically used as a substitute for an SObject when binding to data to a Visualforce page
  - Consists primarily of properties used for binding
  - Typically does not include any methods
- Allows for:
  - Storage of temporary variables
  - Circumvention of license and security restrictions
- Usage Scenarios:
  - Bind a Visualforce page to a wrapper class to allow a User to select multiple records using a checkbox. The value of the checkbox will never exist in the database and is only used as a temporary variable to identify the selected records
  - Bind a Visualforce chart to data that doesn't come directly from a SOQL query

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

AJAX in Visualforce

STONYPOINT

- Visualforce allows developers to create AJAX behavior without writing any Javascript utilizing action tags
- Key tags:
  - <apex:actionSupport />
    - Allows a developer to consume a Javascript event on any component and call an action in the controller
  - <apex:actionStatus />
    - Automatically displays during an Ajax event and hides upon completion of the event
  - <apex:actionPoller />
    - Allows a developer to call an action in the controller on a repeating, delayed basis
  - <apex:commandButton />
    - Allows a user to click a button and call and action in the controller
    - Use of the Rerender attribute causes a partial page refresh
  - <apex:commandLink />
    - Same behavior as commandButton but with a link
  - <apex:outputPanel />
    - The most reliable tag to refresh using the rerender attribute

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Walkthrough: Visualforce Charting

STONYPOINT

Goal:

Create a Visualforce page and embed a Chart on it

Action Steps:

1. Create a new Visualforce page called OpportunityChart
2. Build an Apex class called OpportunityChart to use as the controller for the page
3. Build a wrapper class called ChartData within the OpportunityChart class
4. Add a SelectOption method for Years
5. Add a method to get the Opportunities for the selected year and expose the results as a list of ChartData objects
6. Build an action method called Reload to repopulate the Opportunities
7. Bind the VF page to the OpportunityChart controller
8. Add a SelectList to display the years
9. Add a PieChart to the page and bind it to the ChartData list
10. Add a DataTable and bind it to the ChartData list
11. Add an actionsupport to the onchange event of the select list

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Walkthrough: Visualforce in Dashboards

STONYPOINT

Goal:

Create a Dashboard and add a Visualforce page to it

Action Steps:

1. Create a new Dashboard folder called GenWatt Dashboards
2. Add the Visualforce page OpportunityChart to the Dashboard
3. Save the Dashboard to the GenWatt Dashboards folder and call it Opportunity Dashboard
4. Run the Dashboard

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Integrating with Salesforce

STONYPOINT

- Salesforce Integration Technologies
  - Salesforce to Salesforce
  - SOAP API's
  - REST API's
- Security, Authorization and Authentication
  - Force.com Security
  - Remote Sites
  - Single Sign-on
  - OAuth2

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Salesforce to Salesforce

STONYPOINT

- Salesforce to Salesforce allows sharing of records between different instances of Salesforce
- Once enabled, a Connections tab becomes available
- A Connection is established via an invitation and acceptance
  - Once a Connection is established, both sides can publish and subscribe to objects
- By default, record sharing is manual
  - An External Sharing related list becomes available for each object being published
- Subscriber must map the fields for the incoming records
  - Object and field names do not need to be identical in both instances of Salesforce
- Apex can be used to automate the publishing of records
  - `PartnerNetworkRecordConnection` class

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Walkthrough: Salesforce to Salesforce

STONYPOINT

Goal:

Enable Salesforce to Salesforce and subscribe to invoices created in the instructor's org

Action Steps:

1. Enable Salesforce to Salesforce
2. Create an Account record for Genwatt and create a Contact record for the instructor
3. Using the Connections tab, send an invitation to the instructor
4. The instructor will accept all your invitations and publish the Invoice object
5. Subscribe to the Invoice object and map the fields
6. Set it to auto accept published records
7. The instructor will create an invoice record, publish it to you and it should automatically appear in your org

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

222

Walkthrough: Automating SF2SF with Apex

STONYPOINT

Goal:

Create a trigger on the Invoice object to automatically publish all Invoices to all Salesforce to Salesforce subscribers

Action Steps:

1. Create a trigger called PublishInvoice on the After Insert and After Update events on the Invoice object
2. Query the database to get a list of all SF2SF connections
3. Loop through the Invoice records and create a new record share for each connection and add it to a list
4. Upsert the list of records

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Remote Sites

STONYPOINT

- Designed by Salesforce to prevent malicious code from sending information remotely
- Before any Apex or Javascript code running on the platform can make an external call, the System Administrator must create a Remote Site for the destination URL
- Remote Sites can be added under the Setup -> Security Controls -> Remote Site Settings menu

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.
224

Single Sign-on

STONYPOINT

- Salesforce offers the following ways to use single sign-on:
  - Federated authentication using Security Assertion Markup Language (SAML)
    - Enabled by default for all organizations.
  - Delegated authentication enables you to integrate Salesforce with an authentication method such as LDAP
    - Must be enabled by Salesforce
    - Allows some users to authenticate with normal user name and password with others use the delegate
  - Authentication providers let your users log in to your Salesforce organization using their login credentials from an external service provider.
    - Salesforce supports any OpenId provider
- Salesforce can also act as an identity provider and use single sign-on to connect to a different service provider.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

SOAP

STONYPOINT

- Simple Object Access Protocol (SOAP)
- Utilizes XML to transport data utilizing the HTTP protocol
- Popular in late 1990's and early 2000's
- Fundamental part of the SOA (Service Oriented Architecture) movement
- Being replaced by REST
- Benefits
  - Standard communication protocol
  - Services are documented using WSDL (Web Services Description Language)
  - WSDL is natively supported by all major development tools
- Challenges
  - Data packages very large, too large for mobile devices on mobile networks
  - Response time is slow
  - Client processing time is slow

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.
226

Force.com SOAP API

STONYPOINT

- aka Force.com Web Services API
- The first web service released by Salesforce
- Most tools (Apex Data Loader, Force.com IDE, etc.) still utilize this API
- Highly functional
- Very secure
  - Uses two factor authentication or Session ID
    - Username & password
    - Security Token
- Over 20 standard functions
  - Data manipulation functions: Insert, Update, Delete, Upsert, etc.
  - Search
  - Query
- Developers can easily build custom SOAP web services to extend functionality

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Salesforce WSDL

STONYPOINT

- Enterprise WSDL
  - Methods and definitions for a specific instance of Salesforce
- Partner WSDL
  - Generic methods that allow a developer to integrate with any instance of Salesforce
- Apex WSDL
  - Methods that allow a developer to manipulate Apex code remotely
- Metadata WSDL
  - Methods that allow a developer to manipulate Metadata remotely
- Tooling WSDL
  - Allows a developer to build or extend tools for developing applications on the Force.com Platform

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

SOAP: Retrieving Data with SOQL

STONYPOINT

- It is possible to make SOQL calls using the SOAP API
- Sample Java Code:

```
// SOQL query to use
String soqlQuery = "SELECT FirstName, LastName FROM
Contact";
// Make the query call and get the query results
QueryResult qr = partnerConnection.query(soqlQuery);
```

- Security is always enforced
- Force.com IDE utilizes the API to retrieve the data

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

SOAP Web Services Syntax

STONYPOINT

- To develop a custom SOAP web service, the developer must create a `global` class
- All methods must be annotated with the `webservice` keyword
- All methods must be static
- Methods cannot return Maps, Sets or Exceptions

```
global with sharing class SOAP_Invoices {

    webservice static List<Invoice__c> GetInvoices(Id AccountId){

        List<Invoice__c> invoices = new List<Invoice__c>();

        for(Invoice__c i : [SELECT Id, Name, Amount__c, Due_Date__c,
            Balance__c,
            Status__c FROM Invoice__c
            WHERE Account__c = :AccountId]) {
            invoices.add(i);
        }
        return invoices;
    }
}
```

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Walkthrough: Custom SOAP Web Service

STONYPOINT

Goal:

Develop a class that returns a list of Invoices when passed an Account Id

Action Steps:

1. Create a global class called SOAP_Invoices
2. Create a static webservice method that accepts an AccountId and returns a List of related Invoice records
3. Generate the WSDL for the class

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Consuming SOAP with Apex

STONYPOINT

- The Force.com platform makes it difficult to consume complex SOAP services with Apex
  - Many standard constructors are not supported by Apex
    - Example: Cannot import the Salesforce Enterprise WSDL
  - Cannot call external web services synchronously from a trigger
    - Must use the @future annotation and call asynchronously
  - Can be consumed synchronously from an Apex class used as a Visualforce controller
- Recommended approach is to use middleware
  - Import WSDL from Salesforce
  - Import WSDL from external service

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

OAuth2

STONYPOINT

- The security protocol utilized by Salesforce to grant access to REST Web Services on the Force.com Platform
- Basic OAuth2 Flow:
  1. Client application sends login information plus unique information to identify the client application
  2. Salesforce authenticates user then returns a token to the client application
  3. The client application saves the token and utilizes that token to login on behalf of the user
     a. The client application never knows the Salesforce username and password
  4. The user may revoke the token at any time and stop client access

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.
233

REST

STONYPOINT

- REST stands for Representational State Transfer
- REST sends and receives text (normally XML or JSON) using HTTP or HTTPS commands
- Natively supported by the Force.com Platform
- A developer can:
  - Consume the built in REST Web Services provided by the platform
  - Develop custom REST Web Services using Apex
  - Consume third-party REST Web Services using Apex
  - Consume custom or standard Force.com REST Web Services using Apex
- Apex provides:
  - Helper classes to parse and render XML data structures.
  - Helper classes to parse and render JSON data structures.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

REST Web Services

STONYPOINT

- The Force.com Platform provides many standard web services
  - Authentication using SessionID or Oauth2
  - Retrieve and reset usernames and passwords
  - Query the database using SOQL
  - Create, update and delete records of every standard and custom object
  - Search the database using SOSL
  - Query the schema and retrieve information about the structure of all standard and custom objects
- Developers can also develop their own custom REST web services to mask complexity from other developers wishing to consume the data

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

REST Annotations

STONYPOINT

- To develop a custom REST web service, the developer must properly annotate a `global` class
- The sample class below retrieves an Account record when passed an Account Id as a parameter using an HttpGet

```
@RestResource(urlMapping='/Account/*')
global with sharing class MyRestResource {
    @HttpGet
    global static Account doGet() {
        RestRequest req = RestContext.request;
        RestResponse res = RestContext.response;
        String accountId = req.requestURI.substring(
                   req.requestURI.lastIndexOf('/') + 1);
        Account result = [SELECT Id, Name, Phone, Website
                   FROM Account WHERE Id = :accountId];
        return result;
    }
}
```

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

REST Annotations

STONYPOINT

- Force.com supports the following REST Annotations
- `@RestResource(urlMapping='/yourUrl')`
  - Used at the class level and exposes the class as a REST resource.
  - The URL mapping is relative to https://instance.salesforce.com/services/apexrest/.
  - A wildcard character (*) may be used
  - The URL mapping is case-sensitive
- The following method annotations all require that the method be declared as `global static`
  - `@HttpDelete`
    - Used to annotate a method in the class that deletes a resource
  - `@HTTPGet`
    - Used to annotate a method in the class that returns a value
    - Also called when the HTTP request uses the HEAD method
  - `@HTTPPatch`
    - Used to annotate a method in the class that updates a record
  - `@HTTPPost`
    - Used to annotate a method in the class that creates a new record
  - `@HTTPPut`
    - Used to annotate a method in the class that performs an upsert

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Workbench

STONYPOINT

- Web-based suite of tools designed for administrators and developers to interact with Salesforce.com organizations via the Force.com APIs.
- Support the Partner, Bulk, Rest, Streaming, Metadata, and Apex APIs
  - Allows users to describe, query, manipulate, and migrate both data and metadata
  - Provides features for testing and troubleshooting the Force.com APIs, such as:
    - Customizable SOAP headers
    - Debug logs for API traffic
    - Backward compatibility testing with previous API version
    - Single sign-on integration within the Salesforce application.
- Written in php
  - Can be access on the web at https://workbench.developerforce.com/
  - Can be installed locally on a php compatible web server

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Business Scenario: REST

STONYPOINT

GenWatt would like to be able to display invoices in their current customer portal which is built using traditional web development technologies.

GenWatt has decided that the best way to display invoices is to continue to store them in Salesforce as the system of record and display them in the portal. Every time a portal user wants to see an invoice, the portal will make a call to Salesforce to retrieve the invoice information. The team has decided that building custom REST Web Services is the correct approach.

The team determined that they will need two web services. One service to retrieve a list of all invoices for a particular customer and another service to retrieve a specific invoice by invoice number.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.
239

Walkthrough: Invoice List Web Service

STONYPOINT

Goal:

Develop a class that returns a list of Invoices when passed an Account Id using HTTP GET and test it using Workbench

Action Steps:

1. Create a global class called REST_Invoices
2. Create a method an annotate it with @HttpGet that returns a list of invoices for a particular Account Id
3. Within Salesforce, find the Account Id of an Account record that has an invoice associated with it
4. Using Workbench, authenticate to Salesforce and call the REST Web Service passing in the Account Id

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Consuming REST with Apex

STONYPOINT

- The Force.com Platform makes it fairly easy to consume REST Web Services
- Apex provides 3 main classes
  - HTTP
  - HTTPRequest
  - HTTPResponse
- Example:

```
Http h = new Http();
HttpRequest req = new HttpRequest();
req.setEndpoint(url);
req.setMethod('GET');
HttpResponse res = h.send(req);
return res.getBody();
```

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

XML & JSON

STONYPOINT

- Apex provides helper classes for creating and parsing both XML & JSON
- XML Classes
  - `XMLStreamWriter` class can create well-formed XML documents
  - `XMLStreamReader` class can read through and parse XML documents
- JSON Classes
  - `JSONGenerator` class can generate JSON content.
  - `JSONParser` class can parse JSON content.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.
242

Walkthrough: Calling REST with Apex

STONYPOINT

Goal:

Call a REST web services from within Apex

Action Steps:

1. Create a Connected App
2. Create remote sites to allow call outs from Salesforce
   a. https://na17.salesforce.com
   b. https://login.salesforce.com
3. Create an Apex class called ExternalInvoices
   a. Build a method to authenticate using OAuth2
      i. User: spdev205.int@stonyp.com
      ii. Password: Developer1
      iii. Security Token: DsxRwZVVOCG2FZYPvihJT1uYz
      iv. Consumer Key & Consumer Secret from your Connected App
   b. Build a method to call the Invoices REST web service your instructor created
      i. End point: https://na17.salesforce.com/services/apexrest/Invoices/
      ii. Account Id: 001o00000037YJq
4. Test the classes using Execute Anonymous within the Force.com IDE

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Walkthrough: Calling REST with Workbench

STONYPOINT

Goal:

Call the REST web service to retrieve an Invoice using Workbench

Action Steps:

1. Login to Workbench
2. Navigate to the REST Explorer
3. Call the following service:
   /services/data/v32.0/sobjects/Invoice__c/[InvoiceId]

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

JSON Parser

STONYPOINT

- Class within Apex used to parse JSON strings
  - Typically used to parse the response from a REST web service call
- Typical design patter
  - Loop through the JSON string searching for tokens and getting the values from each token
- Key Methods
  - getCurrentToken()
    - Returns the token that the parser currently points to or null if there's no current token.
  - getText()
    - Returns the textual representation of the current token or null if there's no current token.
  - nextToken()
    - Returns the next token or null if the parser has reached the end of the input stream.
  - nextValue()
    - Returns the next token that is a value type or null if the parser has reached the end of the input stream.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Walkthrough: Deserializing JSON

STONYPOINT

Goal:

Call a REST web services from within Apex and deserialize the results using the JSONParser class

Action Steps:

1. Create a class called ExternalInvoice to hold the results of the REST API call
2. Using execute anonymous, call the ExternalInvoices class and assign the response to a string
3. Read the string into a JSONParser object and iterate through the results
4. For each invoice record returned, create a new ExternalInvoice object, populate the field values and add the object to a list of ExternalInvoices
5. Print the list to the debug log.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Walkthrough: Deserializing JSON using SObjects

STONYPOINT

Goal:

Call a REST web services from within Apex and deserialize the results into a list of SObjects

Action Steps:

1. Using execute anonymous, call the ExternalInvoices class and assign the response to a string
2. Read the string into a JSONParser object and iterate through the results
3. For each invoice record returned, create a new ExternalInvoice object, populate the field values and add the object to a list of ExternalInvoices
4. Print the list to the debug log.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Force.com Streaming API

STONYPOINT

- Designed to automatically push Salesforce data to client applications
- Built on the concept of Long Polling
  - Client makes a request to the server, but instead of responding immediately, the server waits up to 110 seconds to respond
  - If the client doesn't receive a response, it makes another request
- Utilizes:
  - Bayeux protocol for transporting asynchronous messages, primarily over HTTP.
  - CometD as the HTTP-based event routing bus that uses an AJAX push technology pattern known as Comet.
- Has a separate API limit from the REST or SOAP APIs
  - Limit can be increased by Salesforce if necessary

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Streaming API Terms

STONYPOINT

- PushTopic
  - A record that you create.
  - The essential element of a PushTopic is the SOQL query.
  - The PushTopic defines a Streaming APIchannel.
- Event
  - The creation, update, delete, or undelete of a record.
  - Each event may trigger a notification.
- Notification
  - A message in response to an event.
  - The notification is sent to a channel to which one or more clients are subscribed.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.
249

How the Streaming API Works

STONYPOINT

- Create a PushTopic by writing a SOQL query including all the records you wish to receive
  - Decide which DML events will trigger a notification to the subscribers
- Clients Subscribe to the PushTopic
  - Whenever a DML event fires and causes a change to a record included in the query, the subscribers receive a notification
  - The client gets all the records that caused the notification
- PushTopics can be created using Workbench or through Apex

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

SOQL Not Supported

STONYPOINT

- Queries without an Id in the selected fields list
- Semi-joins and anti-joins
- Aggregate queries (queries that use AVG, MAX, MIN, and SUM)
- Queries using COUNT, LIMIT, ORDER BY, GROUP BY, NOT, or OFFSET
- Queries returning Formula fields, Compound address or geolocation fields
- Relationships aren't supported, but you can reference an ID
  - Invalid: Select Account.Name from Opportunity
  - Valid: Select AccountId from Opportunity

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Walkthrough: Creating a PushTopic

STONYPOINT

Goal:

Create a PushTopic using Workbench and subscribe to the topic with Workbench

Action Steps:

1.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Email Services

STONYPOINT

- Allow developers to create an Apex class to accept an inbound email and parse it
- Runs in the context of a specified user
  - Allows for debugging of the service
- Uses
  - Great for integration with legacy systems
  - Not great for receiving emails from people due to challenges parsing email text
- All email services must utilize the following syntax

```
global class myHandler implements Messaging.InboundEmailHandler {

    global Messaging.InboundEmailResult handleInboundEmail(Messaging.InboundEmail
        email, Messaging.InboundEnvelope envelope) {
        Messaging.InboundEmailResult result = new
        Messaging.InboundEmailresult();
        return result;
    }
}
```

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Walkthrough: Creating an Email Service

STONYPOINT

Goal:

Create an email service that creates an invoice from an inbound email

Action Steps:

1. Create a class called CreateInvoiceFromEmail to process the email
   a. Make the class implement Messaging.InboundEmailHandler
2. Create the handleInboundEmail method
3. Create a new Email Service and bind it to the CreateInvoiceFromEmail class
4. Create a new email address for the Email Service
   a. Set the address to accept email from any address
5. Parse the body of the email using the split method and create an invoice record
6. Enable debugging
7. Send an email to the services with [AccountId];[OpportunityId];

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Force.com Sites

STONYPOINT

- Technology that allows developers to build public websites on the Force.com Platform using Visualforce and Apex
- Sites can be authenticated or unauthenticated
- Domain will be [custom].force.com
  - Can utilize a CNAME to serve the site under a custom domain
    - Example: salesforce.stonyp.com
- Difficult to develop
  - Requires writing of custom HTML / Visualforce without use of a WYSIWYG editor
  - If styling using Dreamweaver, set editor to generate XHTML
- Each Site has it's own unique security, profile and system user associated with it
  - To debug a site, debug the system user assigned to the profile assigned to the site

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.
255

Business Scenario: Force.com Sites

STONYPOINT

GenWatt has contracted with a lead generation company, Lead Leaders, Inc. that is going to do prospecting for them. Lead Leaders, Inc. has a proprietary microsite building system and will be driving prospects to this system.

The prospects will fill out a form on the microsite. GenWatt would like a Lead record to be created automatically within their instance of Salesforce whenever a prospect fills out the form. GenWatt considered using Web to Lead, but the volume of leads is expected to exceed 500 per day, so GenWatt needs a different solution. Lead Leaders has informed GenWatt that they can post the form directly to a public web page but have no way to use OAuth security with their system.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Technical Scenario: Force.com Sites

STONYPOINT

GenWatt has decided to build a Force.com Site and create a Visualforce page to accept posts from the forms on the Lead Leaders system.

The Visualforce page will create a Lead record in Salesforce when a post is received.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

257

Walkthrough: Creating a Force.com Site

STONYPOINT

Goal:

Create a Force.com Site

Action Steps:

1. Navigate to Setup -> Build -> Develop -> Sites
2. Choose a domain for your site
   a. Recommended domain: [yourlastname]-spdev205
3. Create a new Site with the following properties
   1. Site Label: GenWattAPI
   2. Site Name: GenWattAPI
   3. Default Web Address: GenWattAPI
   4. Active Site Home Page: UnderConstruction
4. Save the site
5. Review the Profile associated with the site by clicking the Public Access Settings button

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Walkthrough: Creating a public REST API

STONYPOINT

Goal:

Create a Visualforce page to accept a form post and create a Lead record

Action Steps:

1. Create a new Apex class called LeadController
   a. Set the class to without sharing
   b. Create an action method in the class called SaveLead()
   c. Add a string property to the class called Request
2. Create a new Visualforce page called REST_Lead and use LeadController as the controller
   a. Set the action attribute in the <apex:page> tag to call the SaveLead action
   b. Set the ShowHeader attribute in the <apex:page> tag = false
   c. Create an <apex:outputText> and bind it to the Request property
3. Add the REST_Lead page to the Force.com Site
4. Modify the SaveLead() method to:
   a. parse the input parameters of Fname, Lname, Email, & Company
   b. Create a new Lead and insert it into the database
   c. Set the Request property to the input parameters
5. To test the page, navigate the the REST_Lead page and pass in the Lname, Fname, Email & Company parameters through the URL

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Business Scenario: CMS

STONYPOINT

The GenWatt marketing department has been asking for some time to be able to quickly create and publish Press Releases on their website. Now the Force.com Sites has been enabled, it will be fairly simple to publish Press Releases created in Salesforce.

GenWatt has decided to create a Press Release object in Salesforce and publish to the website using Force.com Sites.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.
260

Exercise: Press Release Object

STONYPOINT

Goal:

Create a new custom object for Press Releases.

Action Steps:

1. Navigate Setup -> Build -> Create -> Objects
2. Create a new custom object by clicking on the New Custom Object button
3. Set the following options:
   a. Label: Press Release
   b. Plural Label: Press Releases
   c. Object Name: Press_Release
   d. Record Name: Name
   e. Data Type: Text
   f. Allow Reports
   g. Deployed
4. Save the object by clicking the Save button.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.
261

Exercise: Custom Tabs

STONYPOINT

Goal:

Create a Custom Tab for the Press Release object

Action Steps:

1. Navigate to Setup -> Build -> Create -> Tabs
2. Create a new Custom Object Tab with the following settings:
   a. Object: Press Release
   b. Tab Style: Your Choice of icon and colors
3. Set the Tab visibility to Default On for all Profiles
4. Add the Tab to the Marketing App

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Exercise: Press Release Custom Fields

STONYPOINT

Action Steps:

1. Navigate to Setup -> Build -> Create -> Objects
2. Click on the Press Release Object and navigate to the Custom Fields & Relationships section. Create the following new fields:

Type	Label	Name	Details
Date	Publish Date	Publish_Date	Make Visible to all Profiles
Checkbox	Publish	Publish	Unchecked by Default Make Visible to all Profiles
TextArea (Rich)	Body	Body	Length: 32,768 Visible Lines: 25 Make Visible to all Profiles Add to Invoice Page Layout

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Walkthrough: Press Releases page

STONYPOINT

Goal:

Create a Visualforce page to display all published Press Releases

Action Steps:

1. Create a new Apex class called PressReleases
   a. Add a List<Press_Release__c> property to the class called Releases
   b. Create a method call getReleases() to get all published releases
2. Create a new Visualforce page called PressReleases and use PressReleases as the controller
   a. Create an <apex:dataTable> and bind it to the Releases property
   b. Create <apex:column> tags to display the date & name of the link
   c. Make the Name a hyperlink
3. Add the PressReleases page to the Force.com Site
4. Modify the getReleases() method to:
   a. Get all the published press releases with a release date in the past and add them to the list of Press Releases
5. Create at least one Press Release record and publish it
6. To test the page, navigate to the PressReleases page
   a. If no records show, check the security settings of the profile associated with the site

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Walkthrough: Press Release page

STONYPOINT

Goal:

Create a Visualforce page to display a Press Release

Action Steps:

1. Create a new Apex class called PressRelease
   a. Add a Press_Release__c property to the class called Release
   b. Modify the controller to retrieve a press release based on an id being passed in
2. Create a new Visualforce page called PressRelease and use PressRelease as the controller
   a. Create an <apex:outputField> and bind it to the Release.Name field
   b. Create an <apex:outputField> and bind it to the Release.Publish_Date__c field
   c. Create an <apex:outputField> and bind it to the Release.Body__c field
3. Add the PressRelease page to the Force.com Site
4. Modify PressReleases page to:
   a. Display the name as a hyperlink and pass the Id to the PressRelease page
5. To test, navigate to the PressReleases page and click on a link
   1. If the body displays HTML tags, add the escape="true" attribute to the outText

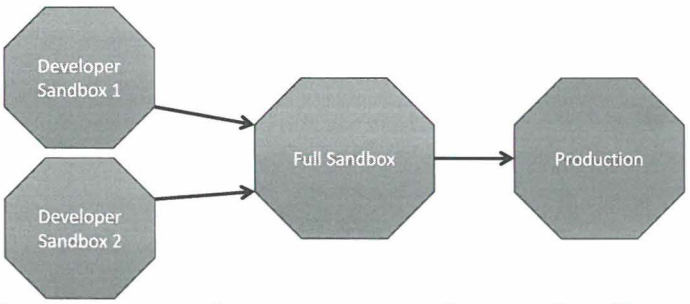
Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.
265

Deployment & Change Management

STONYPOINT

- Deployment - Moving configuration changes from one environment to another
- Change Management – Keeping track of the changes
- Typical enterprise environment



- For more environments, refer to the "**Development Lifecycle Guide: Enterprise Development on the Force.com Platform**" publication on developer.force.com

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012 - 2013 Stony Point, Inc. All rights reserved.

Metadata API

STONYPOINT

- All *configuration* changes move from one Salesforce environment to another using the Metadata API
  - Metadata is data that describes data
  - Examples: Object Definitions, Sharing Rules, Page Layouts, etc…
  - All configuration settings in Salesforce are stored as Metadata
- The vast majority of configuration changes made using the declarative interface are exposed through the Metadata API
  - **Important Note**: Not all configuration settings are available through the Metadata API
    - Examples: Time-based Workflow, Approval Processes
- Cannot use Metadata API to move record data
- Salesforce tools that access the Metadata API directly
  - Force.com IDE
  - Force.com Migration Tool

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012 - 2013 Stony Point, Inc. All rights reserved.

Packages

STONYPOINT

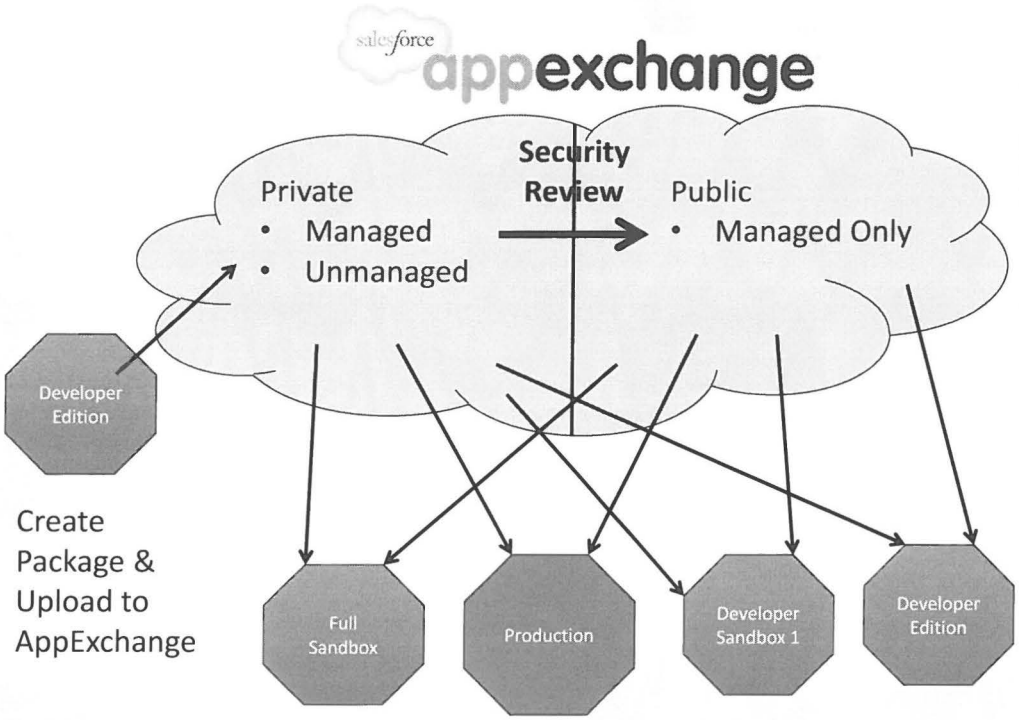
- Similar to an installation package for a desktop application
- Consolidate large amounts of Metadata into a group that can be easily deployed from one Salesforce environment to another
- Two types of packages
  - Unmanaged
    - Installation only
    - Can be changed after installation
  - Managed
    - Install and upgrades
    - Unchanging interface
    - Cannot be changed after installation, only extended
- Use the Metadata API behind the scenes

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012 - 2013 Stony Point, Inc. All rights reserved.

Creating & Distributing a Package

STONYPOINT



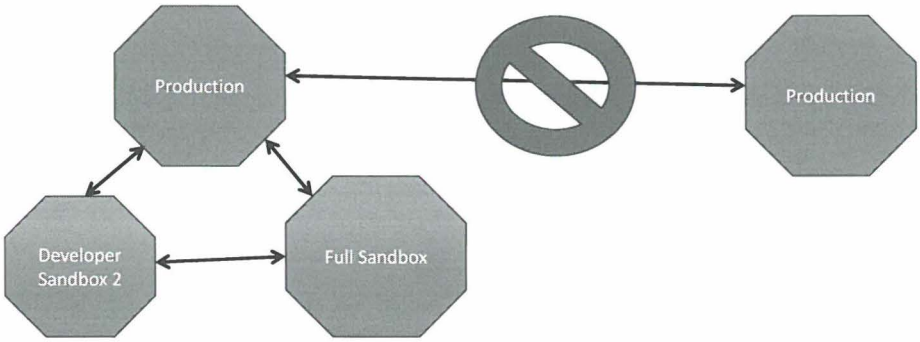
Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012 - 2013 Stony Point, Inc. All rights reserved.

Change Sets

STONYPOINT

- Similar to packages
- Only work within one "family" of Salesforce orgs



- Can be used for install and update
  - Cannot delete or rename
- Use Inbound & Outbound Connections to enforce computer controller rules & change management process
- Use the Metadata API behind the scenes

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012 - 2013 Stony Point, Inc. All rights reserved.

Packages vs. Change Sets

STONYPOINT

	Unmanaged Package	Managed Package	Change Sets
Can be created by	Any type of org	Only a Developer Edition org	Only production or sandbox, not Developer Edition
Can be deployed to	Any type of org	Any type of org	Only orgs in a single family
Modification Types	Install only	Install and update	Install and update
What's included	All dependencies	All dependencies	Only components selected
After install	Can be modified in any way	Can only be extended, not modified	Can be modified in any way
After uninstall	All components are removed and all data is deleted	All components are removed and all data is deleted	Cannot be uninstalled

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012 - 2013 Stony Point, Inc. All rights reserved.

Survey

STONYPOINT

Please take this time to fill out the course survey.

http://schedule.stonyp.com/classsurvey

Your instructor will provide you with the class code to use.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.
272

Question and Answer

STONYPOINT

Please take this time to ask the instructor questions on what you've learned in class.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.

Copyright © 2012-2017 Stony Point, Inc. All rights reserved.