

## to be Creative II

**Martin Giffy D'Souza**

*Martin@ClariFit.com*

*<http://www.TalkApex.com>*

**ClariFit**

[www.ClariFit.com](http://www.ClariFit.com)

- Links:
  - [Martin@ClariFit.com](mailto:Martin@ClariFit.com)
  - <http://www.TalkApex.com>
  - <http://www.ClariFit.com>

# Martin Giffy D'Souza

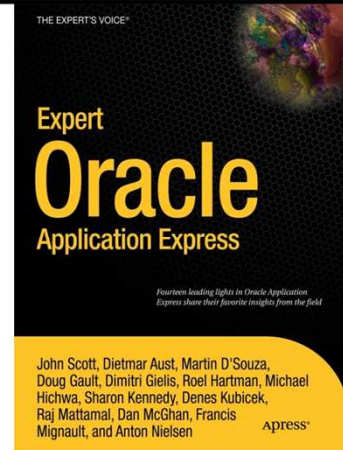
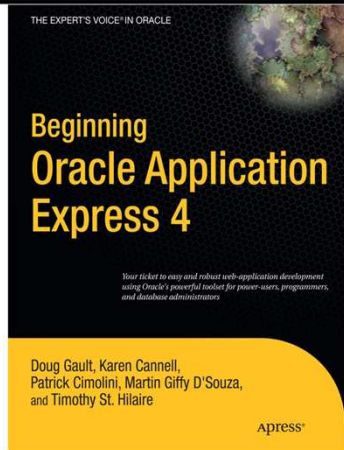


**ORACLE**  
ACE

[www.ClariFit.com](http://www.ClariFit.com)

- Been developing web based applications for over 10 years. Focus primarily on Oracle and APEX over the past 5 years.
- Author of Oracle APEX blog: <http://www.TalkApex.com>
- Recently designated Oracle ACE

# Books

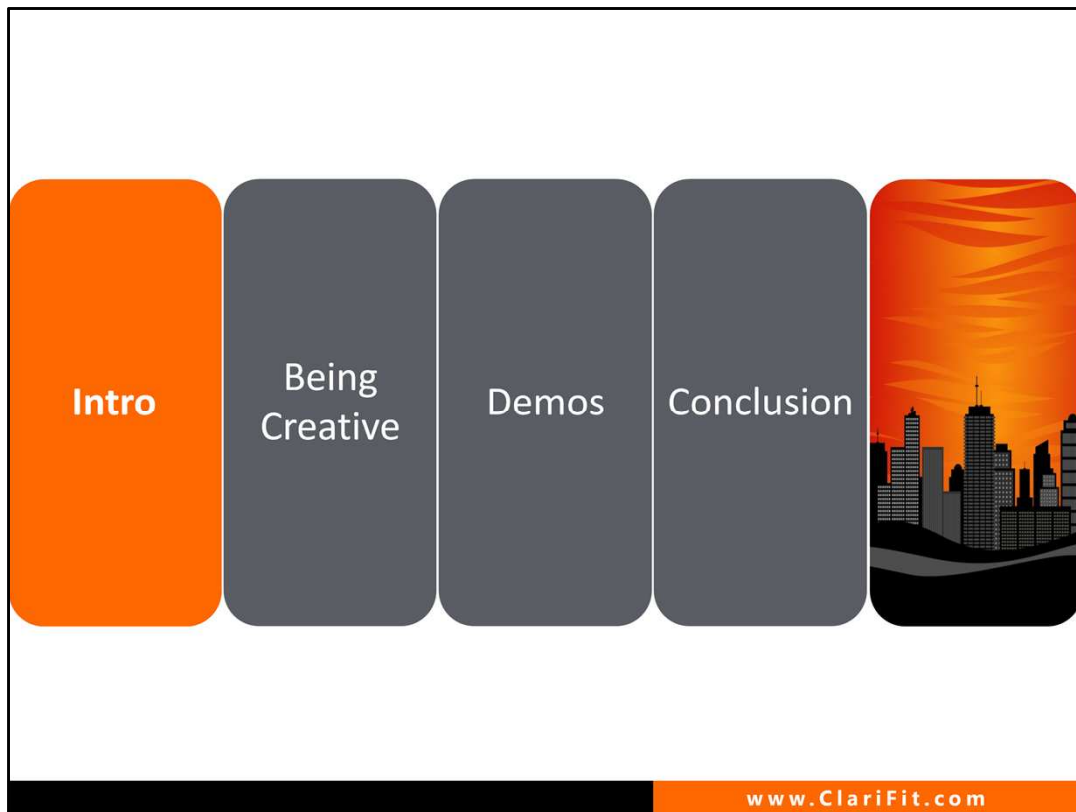


[www.ClariFit.com](http://www.ClariFit.com)

- Co-authored the following books:
  - Beginning Oracle Application Express 4: <http://goo.gl/NxHoF>
  - Expert Oracle Application Express: <http://goo.gl/tXm3P>
    - More info about this book:  
<http://jes.blogs.shellprompt.net/2011/03/30/expert-oracle-application-express>



- ClariFit is a Oracle APEX and PL/SQL consulting and training firm based out of Calgary, Alberta, Canada
- For more information please visit <http://www.ClariFit.com> or email: info@ClariFit.com

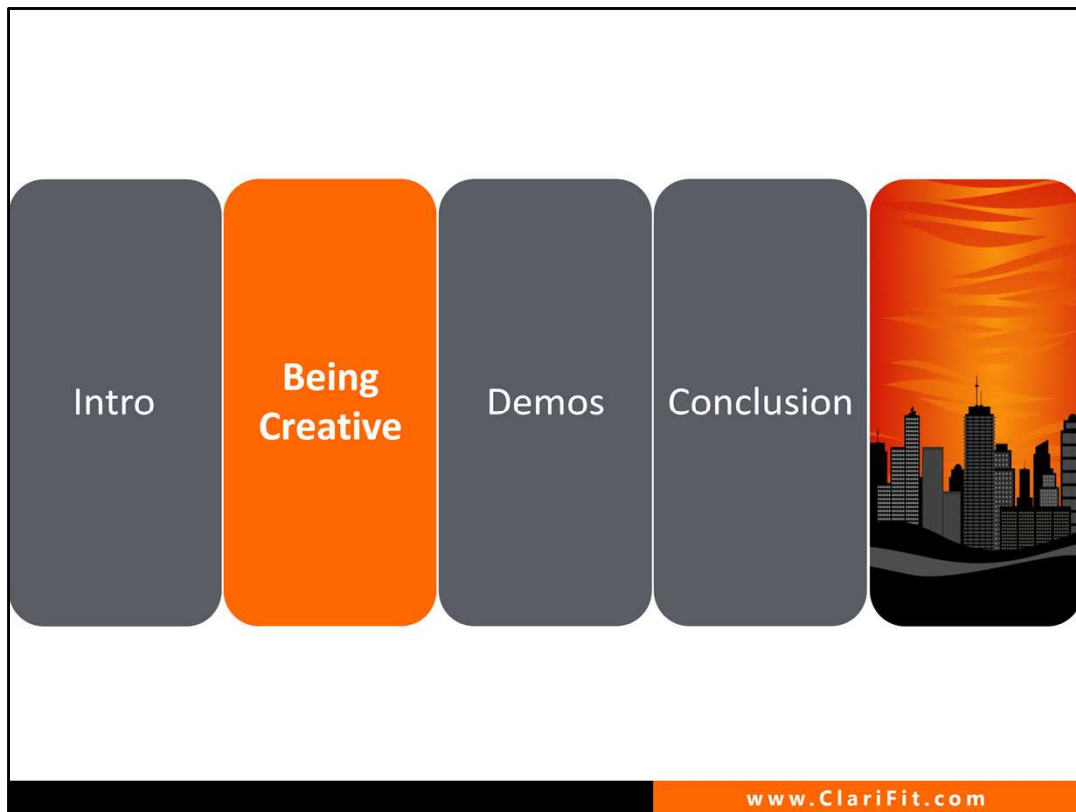


# Assumptions

[www.ClariFit.com](http://www.ClariFit.com)

- Assumptions:

- Have used APEX before
- Have seen or used a bit of JavaScript
  - If you're really new to JavaScript don't worry, you should be able to follow along



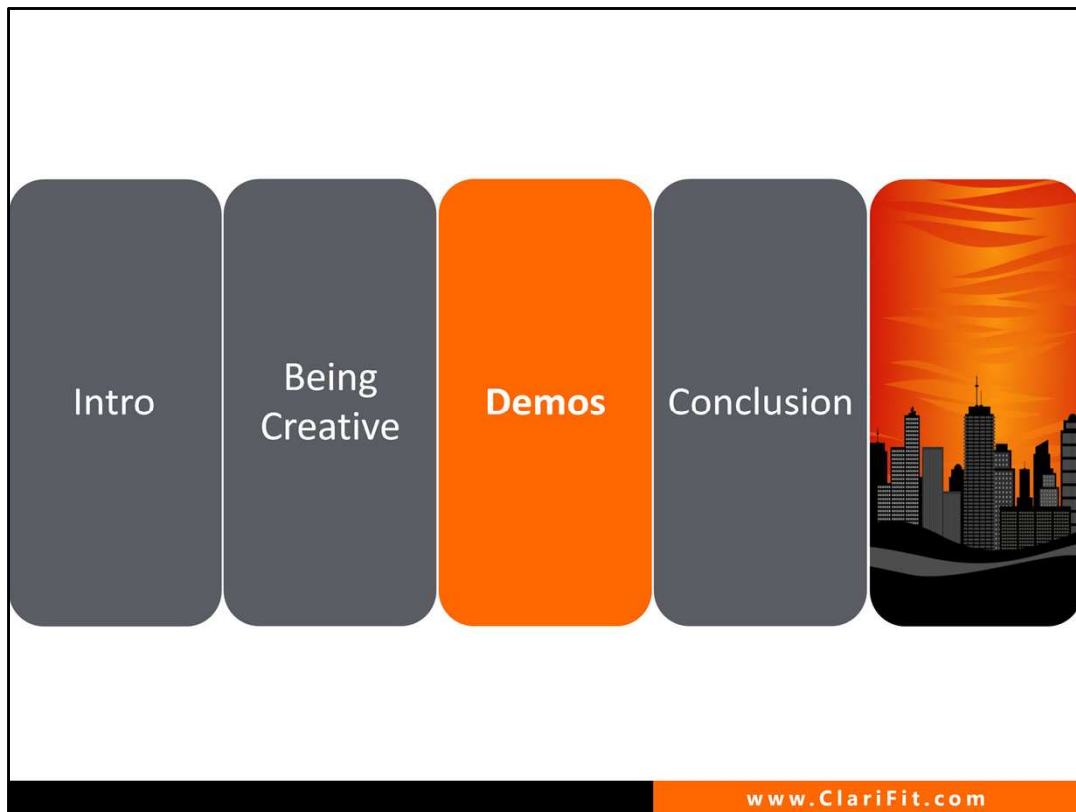
# Being Creative



[www.ClariFit.com](http://www.ClariFit.com)

- Premise behind doing this presentation
  - I've heard many statements such as "APEX Sucks because ...." or "APEX can't do ..."
    - Not the right attitude to take towards the problem
  - APEX is a framework that just produces HTML
    - No different than PHP or .NET
- Yes we face issues with APEX
  - Some tools better for different types of jobs
    - Ex: Interactive Reports has some major benefits over other reporting options in .NET and PHP
  - Need to work around them
    - Get paid to turn business problems into technical solutions
    - This is the difference between programmers and developers





# Code Instrumentation



www.ClariFit.com

- Code Instrumentation
  - Helps with debugging and resolve production issues
  - Tom Kyte is a huge supporter
- In PL/SQL it's really easy to do.
  - Tyler Muth's logger package for PL/SQL code is an excellent tool
    - <https://www.samplecode.oracle.com/sf/projects/logger>
  - APEX\_DEBUG\_MESSAGE for some APEX code
  - [http://download.oracle.com/docs/cd/E17556\\_01/doc/apirefs.40/e15519/apex\\_debug.htm](http://download.oracle.com/docs/cd/E17556_01/doc/apirefs.40/e15519/apex_debug.htm)
- What about JavaScript code?
  - When you write plugins or any JavaScript code to enhance your application they're several ways to do this
  - Can write the standard console.log()
    - Can't easily control when to enable/disable
    - If browser doesn't support (older browsers or disabled) can cause JS errors and crash your app
  - apex.debug
    - JavaScript (not officially supported in the APEX API documentation yet, but will be soon)
    - Will only run when the application is run in debug mode
    - Can only take in a single message.
  - Console Wrapper
    - <http://consolewrapper.clarifit.com>
      - Open source (project hosted on code.google.com)
    - Will run when APEX application is in debug mode
      - or you can manually enable it (see web page for more info)
    - Can leverage most of the features available to console
      - See the "Additional Resources" section on the web page
    - Can take it several messages
    - Supports chaining
    - Auto Log function Params
      - This is a huge time saver
- Demo
  - In a few slides

```
1  function myFn(x, y, z){  
2    var a = x + y;  
3    var b = y + z;  
4    var x = a + b; //Should be a - b  
5    //...  
6    return x;  
7  }
```

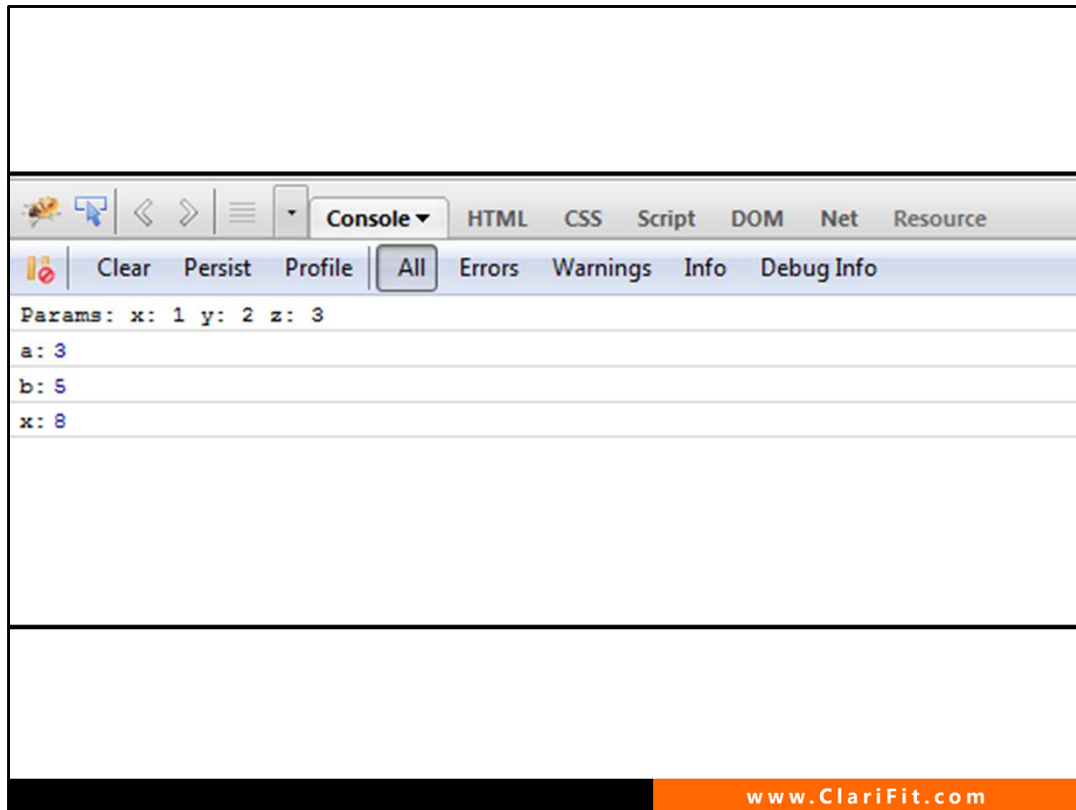
[www.ClariFit.com](http://www.ClariFit.com)

- Suppose you had a bug with your JS code

```
1 | function myFn(x, y, z){
2 |   console.log('Params:' + ' x: ' + x + ' y: ' + y + ' z: ' + z);
3 |   var a = x + y;
4 |   console.log('a:', a);
5 |   var b = y + z;
6 |   console.log('b:', b);
7 |   var x = a + b; //Should be a - b
8 |   console.log('x:', x);
9 |   //...
10 |  return x;
11 | }
```

www.ClariFit.com

- What you would do is instrument the code with *console.log()* calls to help find the issues
  - In this case it's pretty simple code but in more complex JavaScript calls you'll need to see what is going on



- When you run the code, the *console.log* statements will output what is happening
- This information can really help resolve issues quickly

```
1  function myFn(x, y, z){
2    var a = x + y;
3    var b = y + z;
4    var x = a - b; //Fixed
5    //...
6    return x;
7  }
```

www.ClariFit.com

- In the past after resolving the issue what you needed to do was remove all the *console.log* statements since some browsers did not support it and would crash the application
  - Note: The recent version of all major browsers now support it

# Code Instrumentation - Demo

- See notes for demo
- Need to run in FireBug (FireFox) or Console in Chrome/Safari (Ctrl + Shift + J)
- Assumes you have a text item on Page 1 called P1\_X

www.ClariFit.com

```
//You'll need to include the console wrapper JS file in your application somewhere
//Go to: https://consolewrapper.clarifit.com to download it

// Run on Page 1
console.log('hello');
apex.debug('hello');
$.console.log('hello');//This refers to the console wrapper
//Notice how only a "hello" was displayed?

// Run the app in debug mode
console.log('hello');
apex.debug('hello');
$.console.log('hello');
//You should now see all 3 "hello"

//Won't be comparing "console.log()" calls any more

//Now get a bit more advanced
var x = {start: 'hello', end: 'goodbye'};
apex.debug(x);
$.console.log(x);
//Note: both worked and displayed proper JSON object

//What about multiple variables. This can help save some lines
var x = {start: 'hello', end: 'goodbye'};
var y = 123;
apex.debug(x,y);
$.console.log(x,y);

//Only way to use the apex.debug would be for multiple lines
apex.debug(x);
apex.debug(y);

//Going to highlight some more neat features in Console

//Grouping
var x = {start: 'hello', end: 'goodbye'};
var y = 123;
$.console.group('Group Header');
$.console.log('hello');
$.console.log(x, y);
$.console.groupEnd('Group Header');
$.console.log('After group');

//Chaining
$.WFL_REPORT_SEARCH().val('123');
apex.debug('changed value');
apex.debug($.WFL_REPORT_SEARCH());
//Notice how this took 3 lines to write?

//This only takes 1 line
$.WFL_REPORT_SEARCH().val('123').log('changed value');

//Function Arguments
//Suppose you had the following function:
function myFn(x, y, z){
  //...
  return x + y + z;
}

//Using apex.debug you'd need to write out all the parameters manually
//This can take a while
function myFn(x, y, z){
  apex.debug('Parameters: x: ' + x + ', y: ' + y + ', z: ' + z);
  //...
  return x + y + z;
}

//Run The code
myFn('a','b','c');

//Works well but what about complex objects?
myFn('a', 2, {a: 123, c: 456});
//Notice that it doesn't tell you much "just the object"

//What about "extra parameters"
myFn('a', 2, 1, 45);
//Doesn't catch the extra "a"

//Better way?
//Yes there is: the logParams function which is included in the console wrapper
function myFn(x, y, z){
  $.console.logParams();
  //...
  return x + y + z;
}

//These are the same cases as above. Notice how it resolves each of the issues?
myFn('a', 2, 1);
myFn('a', 2, {a: 123, c: 456});
myFn('a', 2, 1, 45);
```

	All Console Fns	Tight APEX Integration	Chaining	Quick Params	Toggle
console.log	✓	No	No	No	No
apex.debug	No	✓	No	No	✓
Console Wrapper	✓	✓	✓	✓	✓
			www.ClariFit.com		

- Summary
- Try to avoid direct calls to console.log
  - Main reason right now is it will always display to the user in their console window
- *apex.debug* is good for basic debugging
  - By no means am I putting it down
  - If you don't want to use the Console Wrapper at lease use *apex.debug*
- Console Wrapper
  - Expands your code instrumentation tool set to
  - Allows you to apply debug code quicker



EMP

Search

SALES

Display

5

Go

Empno	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno
7369	SMITH	CLERK	7902	17-DEC-1980	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-1981	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-1981	1250	500	30
7566	JONES	MANAGER	7839	02-APR-1981	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-1981	1250	1400	30

Spread Sheet

row(s) 1 - 5 of 14

Next

www.ClariFit.com

- Current way to search in standard reports is enter a value then click the “Go” button

<ul style="list-style-type: none"><li>🕒 Page Submitted</li><li>🕒 Validations Run</li><li>🕒 Page Processes</li><li>🕒 Entire Page reloads</li><li>🕒 External Files</li></ul>
<a href="http://www.ClariFit.com">www.ClariFit.com</a>

- Once the page is submitted a lot of things can occur
  - This results in extra server time and time spent by client waiting

EMP

Search

SALES

Display

5

Go

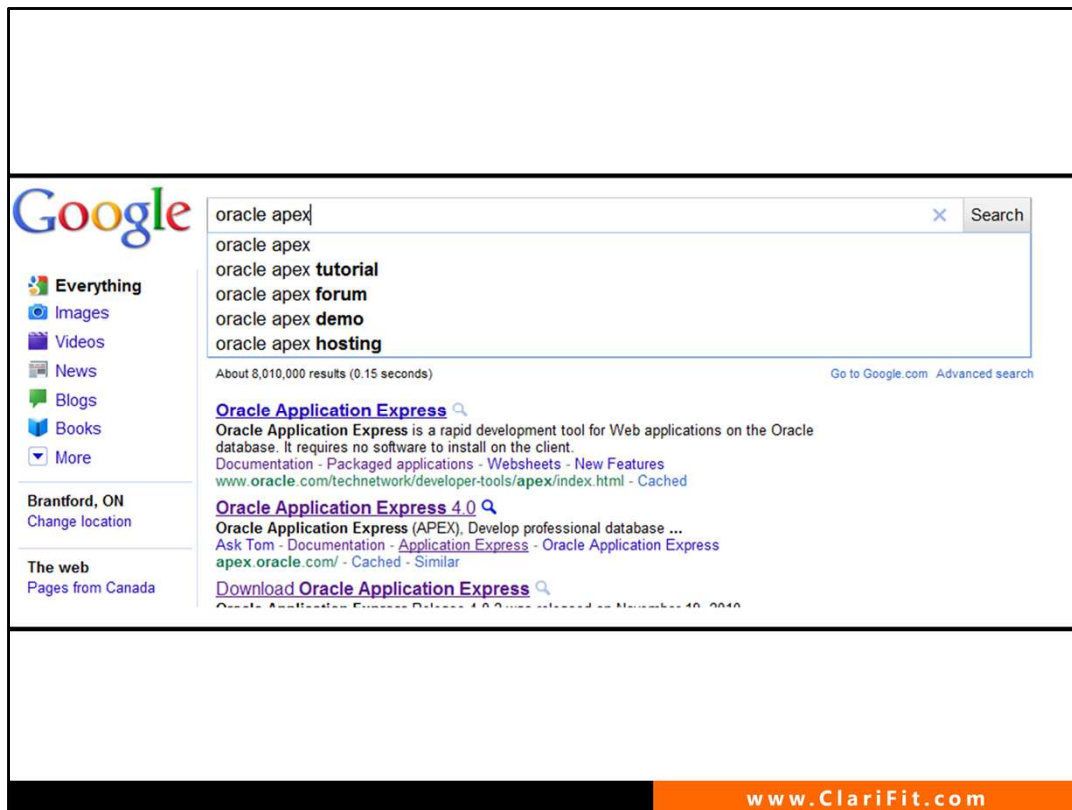
Empno	Ename	Job	Mgr	Hiredate	Sal	Comm	Deptno
7499	ALLEN	SALESMAN	7698	20-FEB-1981	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-1981	1250	500	30
7654	MARTIN	SALESMAN	7698	28-SEP-1981	1250	1400	30
7844	TURNER	SALESMAN	7698	08-SEP-1981	1500	0	30

Spread Sheet

1 - 4

www.ClariFit.com

- You then get your results back
- If you need to refine your search again you have to do it all over again
- Wouldn't it be better if you could see results as you typed?



- Google Instant
  - Search while you type
- Can be helpful if searching for a name and don't know full spelling etc

## Instant Search

- Search results as you type
- For demo go to:  
<http://plugins.clarifit.com> and click on the “Report Instant Search” tab

[www.ClariFit.com](http://www.ClariFit.com)

- Links:
  - <http://plugins.clarifit.com>
- Blog links include installation instructions and detailed background information
- Note: that in APEX 4.1 the plugin will change significantly to leverage the new 4.1 features

# APEX Page Errors



www.ClariFit.com

- Last year talked about APEX page errors
  - Odds are you're getting them but just don't know/realize it
  - More info
    - <http://www.talkapex.com/2010/10/apex-region-errors-part-1.html>
    - <http://www.talkapex.com/2010/10/apex-region-errors-part-2.html>
    - <http://www.talkapex.com/2010/09/custom-error-messages-in-apex.html>
  - Note: The solutions listed above will change dramatically when APEX 4.1 comes out due to it's new way to handle page errors

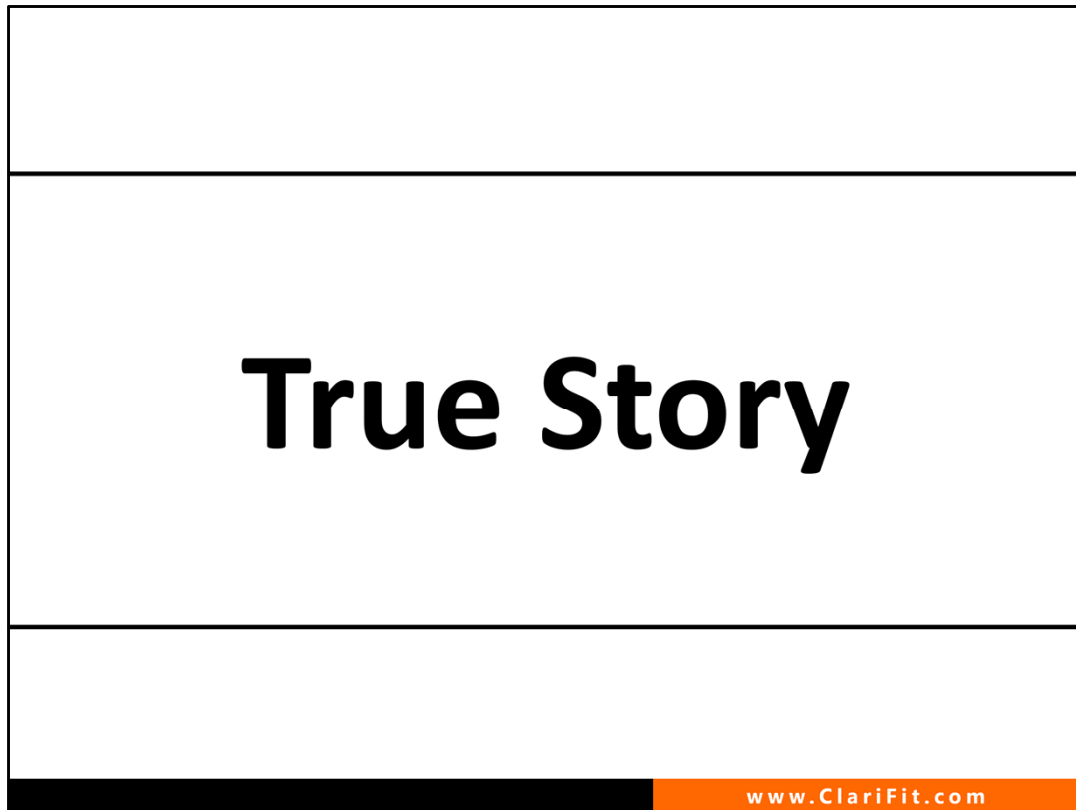
# APEX Region Errors

Emp

```
report error:  
ORA-01476: divisor is equal to zero
```

[www.ClariFit.com](http://www.ClariFit.com)

- What about Region Errors?
- Odds are you have them in your application as well
- Starting in APEX 4.0 there's a way to find out about them



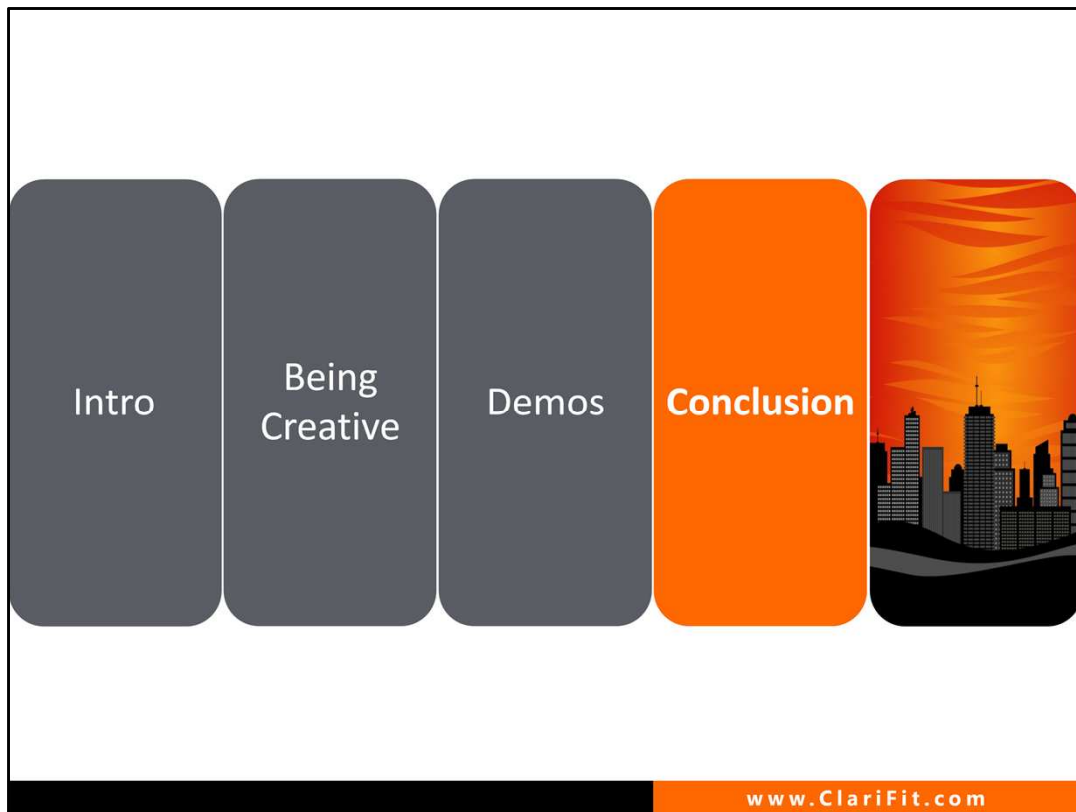
- Story
  - Here is the video (YouTube video) of the story I gave: <http://goo.gl/jmUH8>
  - What it emphasizes is that:
    - When/if users reports a problem you sometimes get very vague information
      - You may not be able to resolve the bug with such little information
    - Worse, is if users try to come up with “magical” solutions themselves which aren’t correct



## APEX Region Errors - Demo

[www.ClariFit.com](http://www.ClariFit.com)

- For demo on how to detect APEX region errors that leverages the FeedBack tool please read:
  - <http://www.talkapex.com/2011/07/apex-region-errors-part-3.html>



- Try to avoid saying “No” the first time you hear a problem
  - There’s usually a simple way to resolve it
- APEX has it’s limitations but by being creative you can get around them

# Martin Giffy D'Souza

---



*Martin@ClariFit.com*

*<http://www.TalkApex.com>*

---



[www.ClariFit.com](http://www.ClariFit.com)

- Links:
  - [Martin@ClariFit.com](mailto:Martin@ClariFit.com)
  - <http://www.TalkApex.com>
  - <http://www.ClariFit.com>