Oracle APEX in action

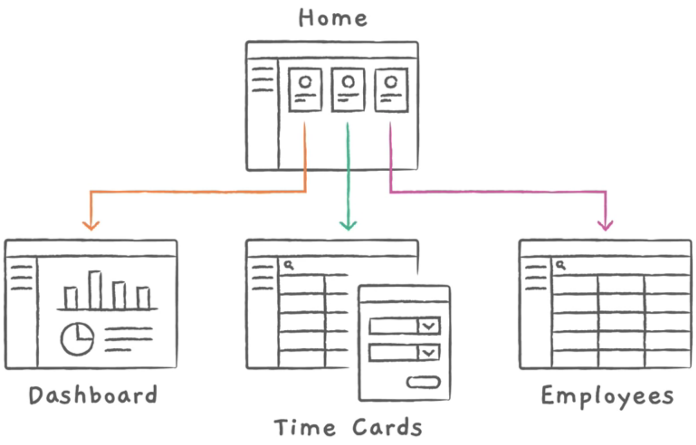
by Novoshore Europe

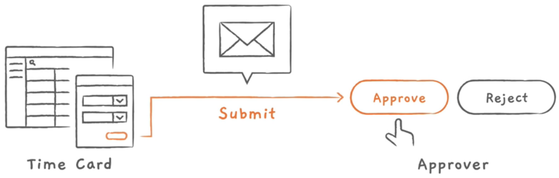
Notes

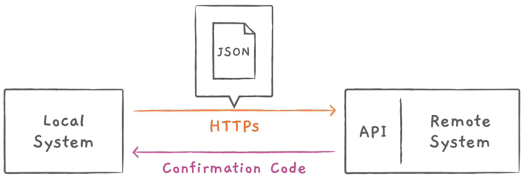
This document is not a formal guide, but a reference guide created to help during a Workshop driven by Novoshore Europe.

If you need any further information about this presentation, please don't hesitate to contact us at any of our contact channels. You can find more details at www.novoshore.com

Timecards APP

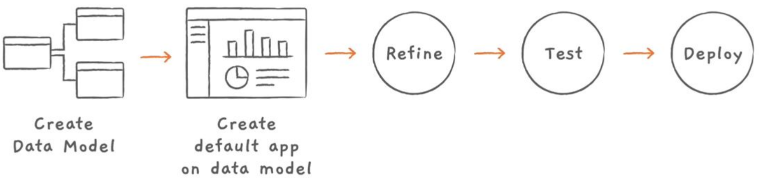






Business rules:

- Hours > 0
- Hours > 20 for part-time employees require justification
- Hours > 40 for full-time employees require justification
- One card per person and week
- Hours might only be entered in arrears



Create Datamodel (QuickSQL): Timecards

```
employees /insert 50 /api /history
        name                            /nn /unique
        email                           /nn /lower /unique
        country                         vc100  /nn /values US, US, US, US, US, Canada, Mexico
        employee_type           vc30    /nn /check full time, part time
        approver                vc255   /nn /index /values tbd
        timecards       /api /auditcols /history /insert 200
                status          /check SUBMITED, APPROVED, DECLINED /default SUBMITTED
                week_of         date    /nn
                sunday          int     /default 0 /between 0 and 24
                monday          int     /default 0 /between 0 and 24
                tuesday         int     /default 0 /between 0 and 24
                wednesday       int     /default 0 /between 0 and 24
                thursday        int     /default 0 /between 0 and 24
                friday          int     /default 0 /between 0 and 24
                saturday        int     /default 0 /between 0 and 24
                comments
                confirmation_code       vc50    /values null

view employee_timecards employees timecards
```

Settings:
- Add row version number
- Add a PL/SQL API
- Include Drops
- History Tables

Save and Run the Script

Errors on drops: that was expected, since those tables don't exist yet

Create the app, using footprint

App Name: Timecards
Add Dashboard:
- Chart 1 -> Pie chart: Employee Type // Employees - Employee_type (label) - Count (value)
- Chart 2 -> Bar chart: Card Status // Timecards - Status (label) - Count (value)
- Chart 3 -> Bar chart: Time Cards By Week // Timecards - Week_of (label) - Coutn (value)

Create and alter order: move Dashboard to second page

Edit History Page: Advanced -> Set as Administrator page
Edit Employees Page: Change icon (fa-users)
Add Features:
- Access Control
- Activity Reporting
- Feedback

Edit Dashboard page

Remove Chart 4 (not in use)
Now, we can run the script listed at the end of the document to display more consistent data

Employees Page (report)

Let's add the Average_Hours that an employee works. For that, let's edit the query:

```
        (select avg(total_hours) from employee_timecards t where t.employee_id = e.id) avg_hours
```

Add a format mask (no decimal)
Run the page

Employee Page (form)

Hide Weekends and commets columns
Remove Employee filter and save as a default report

Add a total hours column to the report:

```
        SUNDAY+MONDAY+TUESDAY+WEDNESDAY+THURSDAY+FRIDAY+SATURDAY
total_hours,
```

Add the employee type:

```
        initcap((select employee_type from employees e where e.id = m.employee_id))
employee_type,
```

Add target hours depending on the employee type:

```
        decode((select employee_type from employees e where e.id = m.employee_id), 'FULL
TIME',40,20) target_hours,
```

Move the Confirmation code to the end
Put Total and Target Hours close to each other
Save report as default again

Refine the app

Edit the timecard page

- Change Employee Label / Confirmation code to the same row
- Set Confirmation code to display only
- Set items together: Status/Week_of - Sun/Wed - Th/Sat (using start new row)
- Add a new item to calculate total hours: P6_TOTAL (display only) - Source Null
- Default value for Sunday: Static value = 0
- Run the page

Create a Validation

Part Time employee Check -> PT employee check

- Error message: Part time employees must provide comments when submitting more than 20 hours
- Validation: PL/SQL Function

```
        if (:P6_SUNDAY + :P6_MONDAY + :P6_TUESDAY + :P6_WEDNESDAY + :P6_THURSDAY
+ :P6_FRIDAY + :P6_SATURDAY) > 20
                and :P6_COMMENTS is null then
                return false;
        else
        return true;
        end if;
```

- Server-side Condition: Rows returned

```
        select 1
```

```
    from employees
        where employee_type = 'PART TIME' and id = :P6_EMPLOYEE_ID
```

Duplicate the validation for Full time employees: FT employee check

- Change the querry to 40
- Change the Error message
- Change server side condition to 'FULL TIME'

Update Timecard form page

Set a default value to all day items to 0
Add a dynamic action to auto calculate the Total Hours item.
Add a select list for Status, and also for Week.

Create a process: Send Email to Approver

```
    for c1 in (select * from employee_timecards where timecard_id = :P6_ID) loop

        apex_mail.send (
            p_to                                => c1.approver,
            p_template_static_id  => 'TIMECARD_APPROVAL',
            p_placeholders                      => '{' ||
            '       "APP_LINK":'                || apex_json.stringify(  '') ||
            '       ,"COMMENTS":'                       ||
apex_json.stringify( c1.comments ) ||
            '       ,"TIMECARD_USER":'                  || apex_json.stringify( c1.email )
||
            '       ,"TOTAL_HOURS":'            || apex_json.stringify( c1.total_hours ) ||
            '       ,"WEEK":'                           ||
apex_json.stringify( to_char(c1.week_of, 'DD-MON-YYYY') ) ||
            '}' );

    end loop;

    apex_mail.push_queue;
```

Email Templates

Go to Share Components to Email Templates

```
    Template name:              Timecard Aprproval
    Static Identifieer: TIMECARD_APPROVAL
    Email Subject:          Please approve this time card
    HTML FORMAT
    Header
        <b style="font-size: 24px;">Time Cards</b>
    Body
        <b>Hello,</b><br>
        <br>
        #TIMECARD_USER# has submitted a time card. Please take a momment to review
        <br><br>
        <table width="100%">
            <tr>
                <th align="left">Week</th>
                <td>#WEEK#</td>
            </tr>
            <tr>
				<th align="left">Total Hours</th>
				<td>#TOTAL_HOURS#</td>
			</tr>
			<tr>
				<th align="left" valign="top">Comments</th>
				<td>#COMMENTS#</td>
			</tr>
		</table>
```

Create a new Authorization Scheme

From Scratch

Name:	Manager
Scheme Type:	Exists SQL Query
SQL Query:	select 1 from employees where approver = lower(:app_user)
Error Message:	You are not authorized to view this page

Edit Page navigation List of page 1. Add the new Auth Schema to Dashboard and Employees
Edit Navigation Menu to do the same thing -> Shared Components > Navigation menu > modify entries.
Edit the different pages to add that restriction at level page (security > Auth schema)

Login as a normal user and go to timecards. Something wrong? I can see all employees. Add some feedback to report that also needs access control on that page.
Login as an admin: Can see everything

Now, modify the Timecard report adding a where clausule:
where employee_id in (select id from employees where approver = lower(:APP_USER) or email = lower(:APP_USER))
That filter options only displays my own records, or records I can Approve.

Verify Security

Go to Utilities > APEX Advisor.
Check Security boxes
As a result: Page 3: missing auth. View > Security > Auth Sche: Manager
Run it again and verify

Explore the APIs we've exposed

Go to SQL Commands and run:

```
begin
employees_api.insert_row (
	p_id			=> null,
	p_name				=> 'JT Thomas',
	p_email				=> 'jt.thomas@novoshore.comm',
	p_country		=> 'US',
	p_employee_type		=> 'FULL TIME',
	p_approver		=> 'juan@novoshore.com'
	);
end;
/
```

Statement processed. 1 row(s) inserted on employees table.

Let's insert some data for the user too:

```
        begin
                for c1 in (select id from employees where email = 'jt.thomas@novoshore.com') loop
                        for c2 in (
                                select next_day(trunc(sysdate), 'sun') - (7 * rownum) week_of
                                from dual
                                connect by level <= 10) loop
                        timecards_api.insert_row(
                                p_id                            => null,
                                p_employee_id                   => cl.id,
                                p_status                        => 'SUBMITTED',
                                p_week_of                       => c2.week_of,
                                p_sunday                        => 0,
                                p_monday                        => 6 + dbms_random.value(1,5),
                                p_tuesday                       => 6 + dbms_random.value(1,5),
                                p_wednesday                     => 6 + dbms_random.value(1,5),
                                p_thursday                      => 6 + dbms_random.value(1,5),
                                p_friday                        => 6 + dbms_random.value(1,5),
                                p_saturday                      => 0,
                                p_comments                      => null,
                                p_confirmation_code             => null
                                );
                        end loop;
                end loop;
        end;
        /
```

This statement inserts some cards with random data. It might lead to an error, due to business rules (validation) that we defined before. If so, modify the null value of p_comments to => 'testing'

Now, I can go to the app, and find this employee (filter JT). If I go inside, I can see some time cards from him.

Procedure to set Config Code

```
create or replace procedure set_conf_code (
        p_timecard_id in varchar2)

        -- begin
        -- for c1 in (select id from timecards where status = 'SUBMITED') loop
        --      set_conf_code(c1.id);
        --      exit;
        -- end loop;
        -- end;

as
        l_rest_url              varchar2(4000) := 'https://apex.oracle.com/pls/apex/timecards/
timecard/approval/';
        l_sql                   varchar2(4000) := 'select timecard_id as "timecard_id",'||
                                                                              '
employee_id as "employee_id",'||
                                                                              '            week_of
as "weeek_of",'||
                                                                              'from
employee_timecards '||
                                                                              'where timecard_id
= :b1';
        l_cursor                sys_refcursor;
```

```
    l_request_body         varchar2(4000);
    l_response             varchar2(4000);
    l_conf_code            varchar2(255);
    l_status               varchar2(255);
    procedure log_status (p_string in varchar2 default null)
    is
    begin
        dbms_output.put_line(p_string);
        --apex_debug.message(p_message=>p_string, p_force=>true);
    end;

begin
    log_status('Start get_conf_code for timecard '||p_timecard_id);
    --
    -- Construct JSON document to be posted to external system
    --
    open l_cursor for l_sql using p_timecard_id;
    apex_json.initialize_clob_output;
    apex_json.open_object;
    apex_json.write( 'timecard', l_cursor );
    apex_json.close_object;
    l_request_body := apex_json.get_clob_output;
    log_status('REST Request: '||l_request_body);
    apex_json.free_output;
    --
    -- Post request to external system
    --
    l_response := apex_web_service.make_rest_request(
        p_url => l_rest_url,
        p_http_method => 'POST',
        p_body => l_request_body );
    log_status('REST Response: '||l_response);
    --
    -- Extract confirmation code from JSON response
    --
    apex_json.parse( l_response );
    l_conf_code := apex_json.get_varchar2( p_path => 'conf_code' );
    l_status    := apex_json.get_varchar2( p_path => 'status' );
    --
    -- Update local database with confirmation code
    --
    update timecards set CONFIRMATION_CODE = l_conf_code, status = l_status where id =
p_timecard_id;
    commit;
    log_status('End set_conf_code for timecard '||p_timecard_id||', conf code: '||l_conf_code||',
status: '||l_status);

end;
```

Run the loop commented at the begining of the procedure and reviw the output.

Export the app

Go to the Export section, and export the app

Deploy it into a different environment

At a second workspace, import the app using the Import section

Adjusting generated data to be more realistic

Go to SQL Workshop
Run SQL Script
Setting all dates to Sunday
Adjusting hours for full vs part time employees
Setting real names to test emails

```
declare
    x int;
begin
--set week_of to sunday
for c1 in (select ID from employees) loop
    x := 1;
    for c2 in (select id from timecards where employee_id = c1.ID) loop
        x := x + 1;
        update timecards set week_of = next_day(trunc(sysdate),'sun') - (7 * x) where id = c2.id;
    end loop;
end loop;
commit;
end;
/

begin
-- fix dixtributions of data to make the data look more realistic
update employees set employee_type = 'FULL TIME' where id in (select id from employees where rownum < 5);
update timecards set sunday = 0, saturday = 0 where employee_id not in (select id from employees where rownum < 10 );
update timecards set status = 'APPROVED' where employee_id not in (select id from employees where rownum < 20);
update timecards set status = 'SUBMITTED' where emplolyee_id in (select id from employees where rownum < 10);
update timecards set monday = round(monday/2) where monday > 3 and employee_id in (select id from employees where employee_type = 'PART TIME');
update timecards set tuesday = round(tuesday/2) where tuesday > 3 and employee_id in (select id from employees where employee_type = 'PART TIME');
update timecards set wednesday = round(wednesday/2) where wednesday > 3 and employee_id in (select id from employees where employee_type = 'PART TIME');
update timecards set thursday = round(thursday/2) where thursday > 3 and employee_id in (select id from employees where employee_type = 'PART TIME');
update timecards set friday = round(friday/2) where friday > 3 and employee_id in (select id from employees where employee_type = 'PART TIME');
update timecards set sunday = sunday - 10 where sunday > 11;
update timecards set monday = monday - 10 where monday > 11;
update timecards set tuesday = tuesday - 10 where tuesday > 11;
update timecards set wednesday = wednesday - 10 where wednesday > 11;
update timecards set thursday = thursday - 10 where thursday > 11;
update timecards set friday = friday - 10 where friday > 11;
update timecards set saturday = saturday - 10 where saturday > 11;
end;
/

begin
--
-- Set managers to be known users in the generated data
--
update employees set approver = 'juan@novoshore.com';
```

```
    update employees set name = 'Juan Lopez', email = 'juan@novoshore.com' where id =
(select min(id) from employees);
    update employees set name = 'Adrian Gomez', email = 'adrian@novoshore.com' where id
= (select min(id) from employees where email not like '%novoshore%');
    end;
    /
```

Adjust the trigger when inserting data into a timecard to enforce the bussines logic

```
-- Trigger Timecards_biu;
-- bellow the line :new.updated_by := nvl (sys_context('APEX$SESSION','APP_USER'),user);

    --
    -- additional code provided to enforce business logic
    --
    if trim(upper(to_char(:new.week_of, 'DAY'))) != 'SUNDAY' then
        RAISE_APPLICATION_ERROR (-20000,'Day of week for timecard must be a
Sunday, date provided: '||to_char(:new.week_of,'Day DD Month YYYY'));
    end if;
    if  nvl(:new.SUNDAY,0)+
        nvl(:new.MONDAY,0)+
        nvl(:new.TUESDAY,0)+
        nvl(:new.WEDNESDAY,0)+
        nvl(:new.THURSDAY,0)+
        nvl(:new.FRIDAY,0)+
        nvl(:new.SATURDAY,0) = 0 then
        RAISE_APPLICATION_ERROR (-20000,'Hours reported must be greater than zero');
    end if;

    for c1 in (select employee_type from employees where id = :new.employee_id) loop
        if  c1.employee_type = 'PART TIME' and
            nvl(:new.SUNDAY,0)+
            nvl(:new.MONDAY,0)+
            nvl(:new.TUESDAY,0)+
            nvl(:new.WEDNESDAY,0)+
            nvl(:new.THURSDAY,0)+
            nvl(:new.FRIDAY,0)+
            nvl(:new.SATURDAY,0) > 20 and
            :new.comments is null then
            RAISE_APPLICATION_ERROR (-20000,'Part time employees must provide
comments when entering more than 20 hours');
        end if;
        if  c1.employee_type = 'FULL TIME' and
            nvl(:new.SUNDAY,0)+
            nvl(:new.MONDAY,0)+
            nvl(:new.TUESDAY,0)+
            nvl(:new.WEDNESDAY,0)+
            nvl(:new.THURSDAY,0)+
            nvl(:new.FRIDAY,0)+
            nvl(:new.SATURDAY,0) > 40 and
            :new.comments is null then
            RAISE_APPLICATION_ERROR (-20000,'Full time employees must provide
comments when entering more than 40 hours');
        end if;
    end loop;

--end timecards_biu;
```

Customise the Login Page

Add static files to the application: Backgroun image (login-bg.jpg) / Login Logo Image (logo-login.png) / Logo image (logo.png)

Add css to the Inline section of the login page:

```
	.t-Body {
		background-image: url('#APP_IMAGES#login-bg.png');
		background-size: cover;
		background-position: 50%;
	}

	.app-icon {
		background-image: url(#APP_IMAGES#logo-login.png); //????
		width: 320px;
		height: 96px;
		background-size: cover;
		background-color: transparent;
	};
```

Adding the Dynamic Action that adds all the daily values to Total

- Calculate Total
- On change of hours
- Set Value: PL/SQL Expression

```
	:P6_SUNDAY + :P6_MONDAY + :P6_TUESDAY + :P6_WEDNESDAY + :P6_THURSDAY
+ :P6_FRIDAY + :P6_SATURDAY
```

Add metatags to the App definition

Add static files to the app, in Shared Components > Static Application Files.
Navigate to Shared Components > User intervace
Include the following code to Favicon HTML section:

```
	<link rel="shortcut icon" href="#APP_IMAGES#favicon/180.ico">
	<link rel="icon" sizes="16x16" href="#APP_IMAGES#favicon/16.png">
	<link rel="icon" sizes="32x32" href="#APP_IMAGES#favicon/32.png">
	<link rel="apple-touch-icon" sizes="180x180" href="#APP_IMAGES#favicon/180.png">
	<meta name="apple-mobile-web-app-capable" content="yes">
	<meta name="mobile-web-app-capable" content="yes">
```