

27.03.2026

# You've Got Mail!

Email Sending Made Easy in APEX

Timo Herwix, Senior Consultant @ Hyand Solutions

code of change

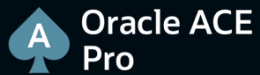


# Who am I?



Timo Herwix

Senior Consultant



Senior Consultant at Hyand Solutions since 2019

Previously worked as a Data Warehouse Developer

Oracle APEX since 2016

Oracle Databases since 2008

Blog author, conference speaker

Born in 1983, two children and living in Germany



# Internationally connected delivery power.



## Germany

- Brunswick
- Ratingen
- Hamburg
- Dortmund
- Frankfurt
- Munich
- Berlin

## Poland

- Wrocław

## Lithuania

- Vilnius
- Kaunas

## Romania

- Cluj-Napoca

## India

- Pune

25+

Customers with  
> 1 billion €  
revenue

110+

Million Euros  
revenue  
(2024)

14

Locations in  
5 countries

750+

Employees

24

Nationalities

1

Basic knowledge  
you should know

2

Advanced knowledge  
you can know

3

Ready to take off

# Pre-Configuration

# Pre-Configuration

**By default, Oracle Database blocks network access for security reasons**

*To send emails from Oracle APEX, network services must be enabled first*

Required Step:

- ✓ Grant network access via DBMS\_NETWORK\_ACL\_ADMIN
- ✓ Allow connect privilege for APEX schema (e.g. APEX\_240200)

Why this is necessary:

- ✉ Enables outbound email (SMTP)

Without this configuration:

- ✗ Email sending and external network calls will fail

```
1 BEGIN
2     DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE(
3         host => '*',
4         ace => xs$ace_type(privilege_list => xs$name_list('connect'),
5                             principal_name => APEX_APPLICATION.g_flow_schema_owner,
6                             principal_type => xs_acl.ptype_db));
7 END;
8 /
```

# Pre-Configuration

To enable APEX to send mail, an instance administrator must configure instance-level email settings.

**Instance Settings** Cancel Apply Changes

Show All Workspace Creation Email Provisioning Storage **Email** Wallet Report Printing Application Gallery Workflow Settings Help New Workspace Request Size

### Email

Oracle APEX Instance URL  ?  
Default URL

Oracle APEX Images URL  ?  
Default URL

**SMTP Host Address**  ?

**SMTP Host Port**  ?

SMTP Authentication Username  ?

SMTP Password  ?

Confirm SMTP Password  ?

Use SSL/TLS  ?

Default Email From Address  ?

**Maximum Emails per Workspace**  ?

Maximum Emails Processed per Workspace  ?

# Pre-Configuration

- apex.oracle.com (oracleapex.com)
  - No configuration required
- Using Autonomous AI Database
  - Configure the OCI Email Delivery Service (create email domain, approved sender addresses, generate SMTP credentials, configure your application)

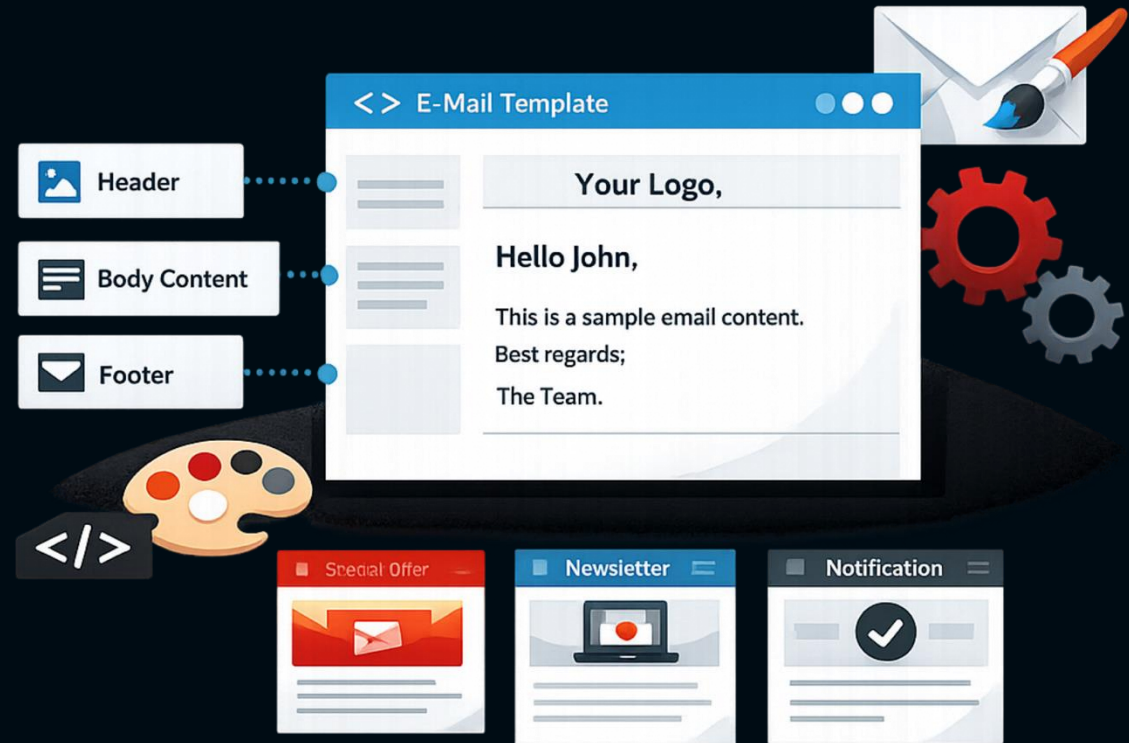
Email Delivery			
Service	Comparison price (/vCPU)*	Unit price	Unit
Oracle Cloud Infrastructure Service - Email Delivery (First 3,000 emails)	—	—	1,000 emails sent
Oracle Cloud Infrastructure Service - Email Delivery (Greater than 3,000 emails)	—	0,07905 €	1,000 emails sent

- Configure a third-party email provider

# E-Mail Templates

# E-Mail Templates

- Introduced in APEX 18.1
- Responsive
- Header, Body and Footer
- HTML and Plain Text Support
- Use of placeholders
- Maintainable
- Reusable



# Sending emails with “E-Mail Templates”

# Sending emails with “E-Mail Templates”

The screenshot shows a 'Settings' dialog box for sending an email. The dialog is dark-themed and contains the following fields and controls:

- From:** &APP\_EMAIL.
- To:** &USER\_EMAIL.
- Cc:** (empty)
- Bcc:** (empty)
- Reply To:** (empty)
- Email Template:** Event Details (dropdown menu)
- Placeholder Values:** No placeholders. (text field)
- Language Override:** (empty)
- Attachment SQL:** (empty text area with a link icon)
- Send Immediately:** A toggle switch that is currently turned on.

At the top of the dialog, there are icons for a list, a lightning bolt, a refresh, and a link. On the left side, there is a vertical menu with options: After Submit, Validating, Processing, Processes (with a sub-menu), Send (highlighted in green), After Processing, and Ajax Callback.

The screenshot shows the 'Template Details' configuration page in a dark-themed interface. The page has a header with 'Cancel', 'Delete', and 'Apply Changes' buttons. Below the header, there are tabs for 'Show All', 'Identification', 'Subscription', 'HTML Format', 'Plain Text Format', 'Comments', and 'Sample API Usage'. The 'Identification' tab is active.

**Identification Section:**

- Template Name:** Event Details
- Static Identifier:** EVENT\_DETAILS
- Email Subject:** Event Reminder: #EVENT\_NAME#
- Supports Template Directives:** Yes

**Subscription Section:**

- Subscribe From:** (dropdown menu)
- Text:** This is the "master" copy of this email template. No email template subscribed to this email template.

**HTML Format Section:**

- Header:** APEX Assistant
- Body:**

```
1 <b style="font-size: 24px;">My Application</b>
```
- Footer:** APEX Assistant
- Footer Content:**

```
1 <a rel="noopener noreferrer" href="#MY_APPLICATION_LINK#">Visit My Application and manage your email preferences</a>.
```

# Sending emails with the “APEX\_MAIL” API

# Sending emails with the “APEX\_MAIL” API

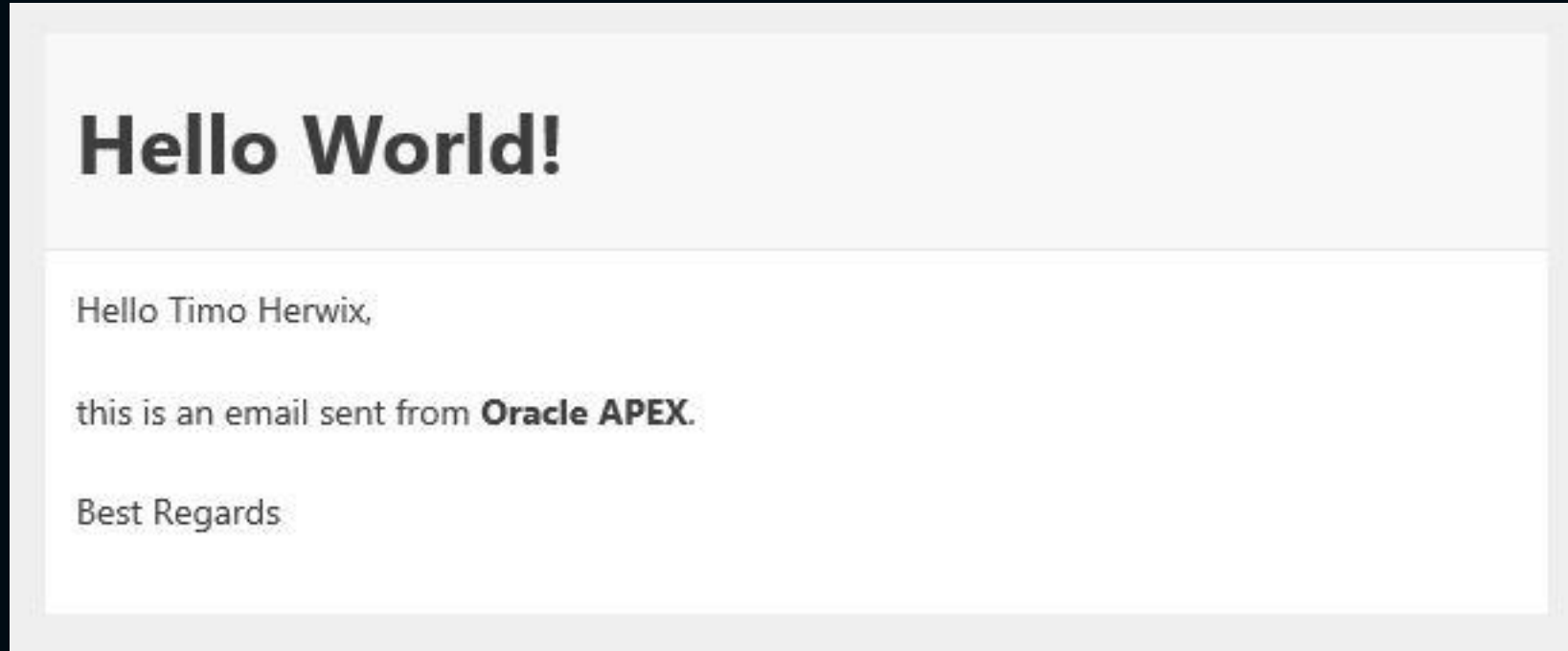
## Sample API Usage

```
begin
  apex_mail.send (
    p_to          => email_address_of_user,
    p_template_static_id => 'HELLO_WORLD',
    p_placeholders => '{' ||
      'CONTACT': ' || apex_json.stringify( some_value ) ||
    '}' );
end;
```

# Sending emails with the “APEX\_MAIL” API

```
1  begin
2    -- This procedure adds a mail to the mail queue of Oracle APEX
3    apex_mail.send(
4      -- Valid email address to which the email is sent
5      p_to          => :P1_TO,
6      -- Email address from which the email is sent
7      p_from        => :APP_EMAIL,
8      -- Static identifier string, used to identify the shared component email template
9      p_template_static_id => 'HELLO_WORLD',
10     -- JSON string representing the placeholder names along with the values, to be substituted
11     p_placeholders => '{' ||
12     '    "CONTACT":' || apex_json.stringify( :CONTACT ) ||
13     '}'
14   );
15
16   -- Oracle APEX stores unsent email messages in a table named APEX_MAIL_QUEUE.
17   -- You can manually deliver mail messages stored in this queue to the
18   -- specified SMTP gateway by invoking the APEX_MAIL.PUSH_QUEUE procedure.
19   apex_mail.push_queue;
20 end;
```

# Sending emails with the “APEX\_MAIL” API



# Sending emails with attachment

# E-Mail Attachment

Settings

From: &APP\_EMAIL.

To: &USER\_EMAIL.

Cc:

Bcc:

Reply To:

Email Template: Event Details

Placeholder Values: No placeholders.

Language Override:

Attachment SQL:

Send Immediately:



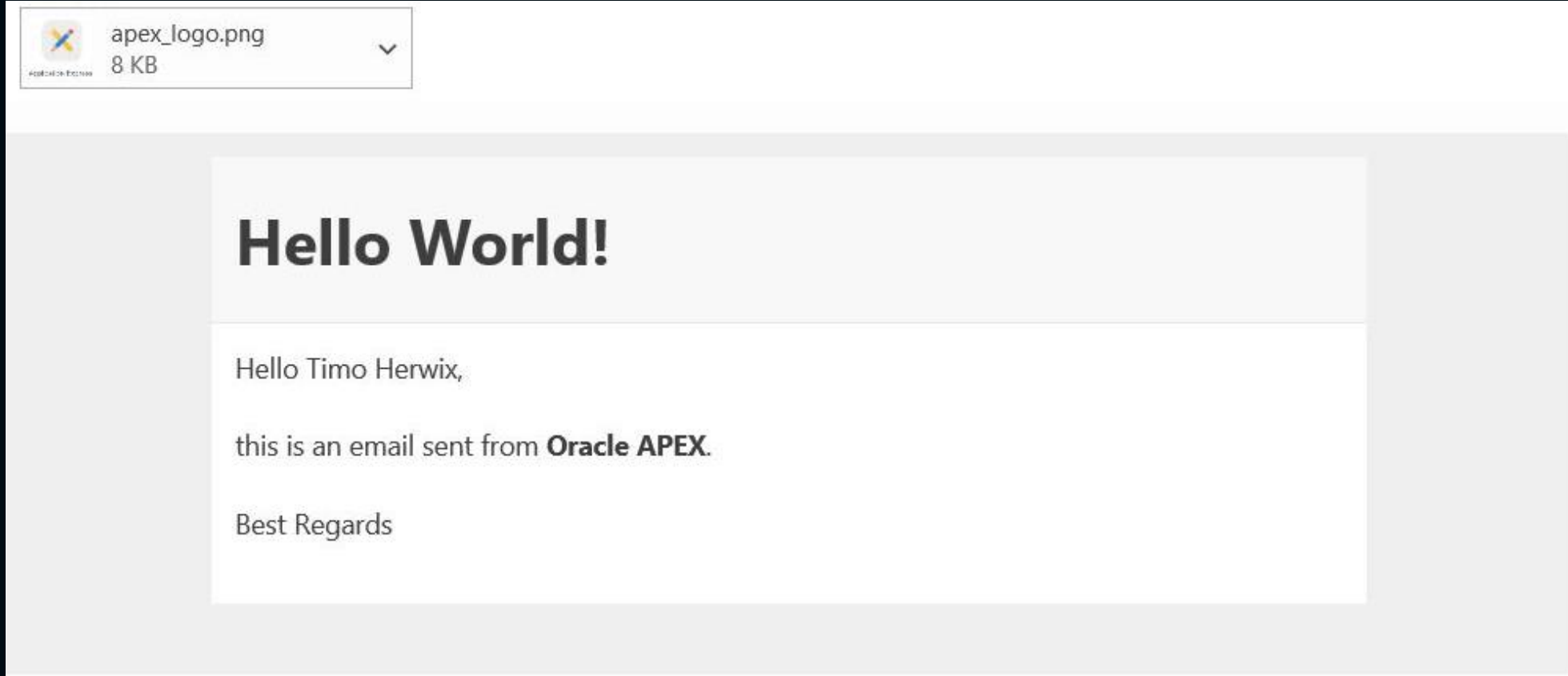
```
1 select blob_content,  
2     filename,  
3     mime_type  
4 from apex_application_files  
5 where flow_id = :APP_ID  
6     and filename = 'apex_logo.png';
```

# E-Mail Attachment



```
1  -- to add multiple attachments to a single email,
2  -- APEX_MAIL.ADD_ATTACHMENT can be called repeatedly in a loop
3  for rec in (
4      select filename, blob_content, mime_type
5          from apex_application_files
6          where flow_id = :APP_ID
7              and filename = 'apex_logo.png'
8  )
9  loop
10     -- this procedure adds an attachment to an email
11     apex_mail.add_attachment(
12         p_mail_id    => l_id,
13         p_attachment => rec.blob_content,
14         p_filename   => rec.filename,
15         p_mime_type  => rec.mime_type);
16 end loop;
```

# E-Mail Attachment



Preview of your email

# Preview of your email

```
1 declare
2     l_placeholders CLOB;
3 begin
4     -- build a JSON formatted string for the placeholders
5     APEX_JSON.initialize_clob_output;
6     APEX_JSON.open_object;
7     APEX_JSON.write('CONTACT', :P2_CONTACT);
8     -- If there are more placeholders
9     -- they can be added here with another APEX_JSON.write
10    APEX_JSON.close_object;
11    l_placeholders := APEX_JSON.get_clob_output;
12    APEX_JSON.free_output;
13
14    -- Preview Email
15    apex_mail.prepare_template (
16        -- Static id string, used to identify the shared component email template
17        p_static_id      => :P2_EMAIL_TEMPLATE
18        -- JSON string representing the placeholder names along with the values
19        ,p_placeholders => l_placeholders
20        -- The subject line generated from the template
21        ,p_subject       => :P2_SUBJECT
22        -- The HTML code for the email
23        ,p_html          => :P2_BODY_HTML
24        -- The plain text of the email
25        ,p_text          => :P2_BODY_TEXT
26    );
27 end;
```

# Preview of your email

Excel Demo demo

**Send Email**

To: t.herwix@googlemail.com

Contact: Timo Herwix

Email Template: Hello World

[Send email](#) [Send email with AP](#)

**Preview**

**E-Mail**

**From:** oracle-application-express\_ww@oracle.com  
**To:** t.herwix@googlemail.com  
**Subject:** Hello World

**HTML Body**

**Hello World!**

Hello Timo Herwix,

this is an email sent from **Oracle APEX**.

Best Regards

**Plain Text**

Hello Timo Herwix, this is an email sent from Oracle APEX. Best Regards

1

Basic knowledge  
you should know

2

Advanced knowledge  
you can know

3

Ready to take off

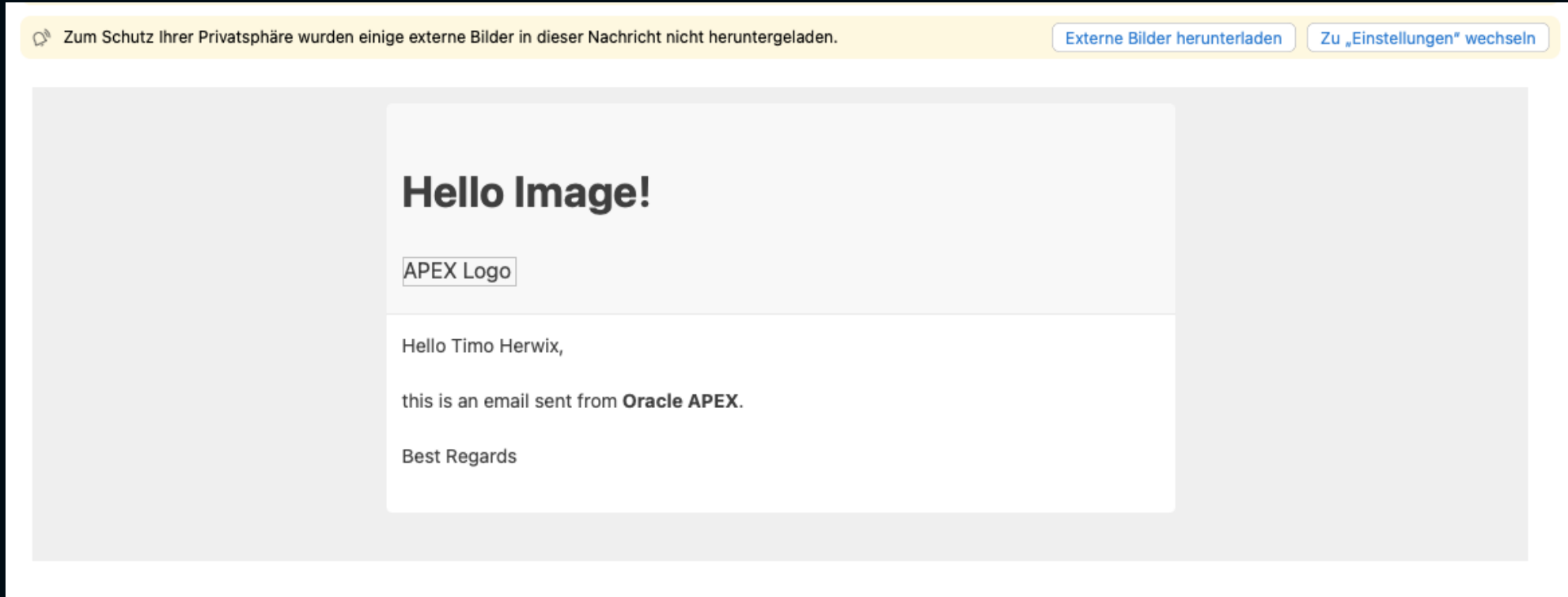
# Embedding images in an email

# Embedding images in an email



```
1 
```

# Embedding images in an email



# Embedding images in an email

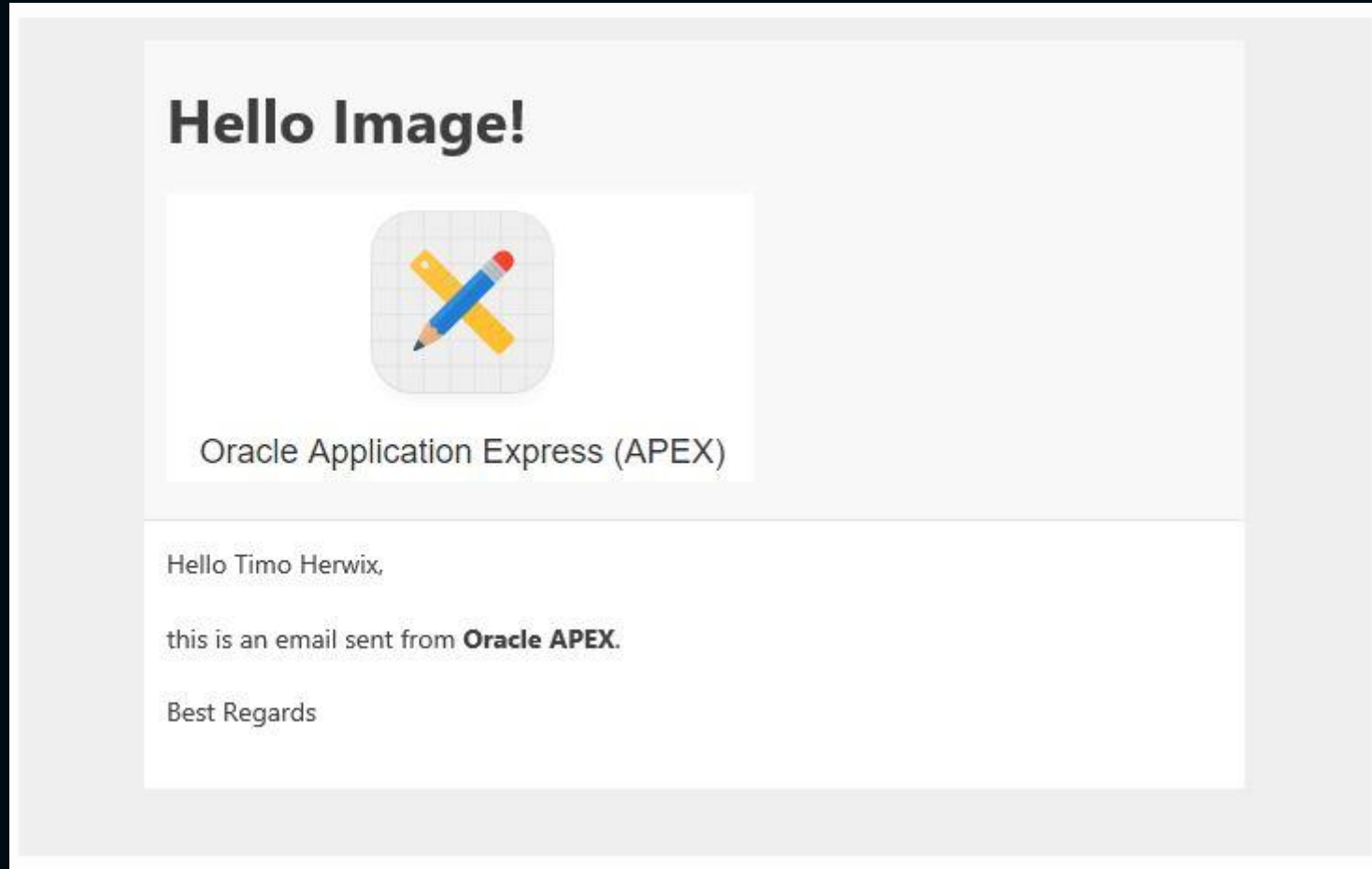


```
1 
```



```
1 select blob_content,  
2        filename,  
3        mime_type,  
4        'apex_logo.png' as content_id  
5 from apex_application_files  
6 where flow_id = :APP_ID  
7        and filename = 'apex_logo.png'
```

# Embedding images in an email



# Display tabular data in an email

# Display tabular data in an email

```
1 declare
2   l_tabular_data VARCHAR2(32767 CHAR);
3 begin
4   -- add table header
5   l_tabular_data := l_tabular_data || '<table>'
6                   || '<tr>'
7                   || '<th>EMPNO</th>'
8                   || '<th>ENAME</th>'
9                   || '<th>DEPTNO</th>'
10                  || '<th>DNAME</th>'
11                  || '</tr>';
12
13   -- add table data
14   for rec in (select empno, ename, deptno, dname from emp_dept_v)
15   loop
16     l_tabular_data := l_tabular_data || '<tr>'
17                       || '<td>' || rec.empno || '</td>'
18                       || '<td>' || rec.ename || '</td>'
19                       || '<td>' || rec.deptno || '</td>'
20                       || '<td>' || rec.dname || '</td>'
21                       || '</tr>';
22   end loop;
23
24   -- close table
25   l_tabular_data := l_tabular_data || '</table>';
26
27   :P1_TABULAR_DATA := l_tabular_data;
28 end;
```

# Display tabular data in an email

**Hello Report!**

Hello Timo,

this is an email sent from **Oracle APEX**.

EMPNO	ENAME	DEPTNO	DNAME
7839	KING	10	ACCOUNTING
7782	CLARK	10	ACCOUNTING
7934	MILLER	10	ACCOUNTING
7566	JONES	20	RESEARCH
7788	SCOTT	20	RESEARCH
7902	FORD	20	RESEARCH
7369	SMITH	20	RESEARCH
7876	ADAMS	20	RESEARCH
7698	BLAKE	30	SALES
7499	ALLEN	30	SALES
7521	WARD	30	SALES
7654	MARTIN	30	SALES
7844	TURNER	30	SALES
7900	JAMES	30	SALES

Best Regards

```
Body ?
APEX Assistant
1 <p>Hello #CONTACT#,</p>
2 <p>this is an email sent from <strong>Oracle APEX</strong>.</p>
3 <br>
4 #REPORT!RAW#
5 <br>
6 <p>Best Regards</p>
```

Using **!RAW** allows us to render pre-generated HTML content such as tables directly inside the email template without escaping

# Checking the mail log and mail queue

# Checking the mail log and mail queue

apex_mail_log		
Column	Data Type	Length
MAIL_ID	NUMBER	22
MAIL_MESSAGE_ID	VARCHAR2	4000
MAIL_TO	VARCHAR2	4000
MAIL_FROM	VARCHAR2	4000
MAIL_REPLYTO	VARCHAR2	4000
MAIL_SUBJ	VARCHAR2	4000
MAIL_CC	VARCHAR2	4000
MAIL_BCC	VARCHAR2	4000
MAIL_MESSAGE_CREATED	DATE	7
MAIL_MESSAGE_SEND_BEGIN	DATE	7
MAIL_MESSAGE_SEND_END	DATE	7
MAIL_BODY_SIZE	NUMBER	22
MAIL_BODY_HTML_SIZE	NUMBER	22
MAIL_ATTACHMENT_COUNT	NUMBER	22
MAIL_ATTACHMENT_SIZE	NUMBER	22
MAIL_SEND_ERROR	VARCHAR2	4000
LAST_UPDATED_ON	DATE	7
APP_ID	NUMBER	22
WORKSPACE_ID	NUMBER	22
WORKSPACE_NAME	VARCHAR2	255

apex_mail_queue		
Column	Data Type	Length
ID	NUMBER	22
MAIL_TO	VARCHAR2	4000
MAIL_FROM	VARCHAR2	4000
MAIL_REPLYTO	VARCHAR2	4000
MAIL_SUBJ	VARCHAR2	4000
MAIL_CC	VARCHAR2	4000
MAIL_BCC	VARCHAR2	4000
MAIL_BODY	CLOB	4000
MAIL_BODY_HTML	CLOB	4000
MAIL_SEND_COUNT	NUMBER	-
MAIL_SEND_ERROR	VARCHAR2	4000
LAST_UPDATED_BY	VARCHAR2	255
LAST_UPDATED_ON	DATE	7
MAIL_MESSAGE_CREATED	DATE	7
MAIL_MESSAGE_CREATED_BY	VARCHAR2	255
APP_ID	NUMBER	22
WORKSPACE_ID	NUMBER	22
WORKSPACE_NAME	VARCHAR2	255

# Checking the mail log and mail queue

Mail Log

Q  Go Actions

Mail Id	Mail Message Id	Mail To	Mail From	Mail Replyto	Mail Subj	Mail Cc	Mail Bcc	Mail Message Created	Mail Message Send Begin	Mail Message Send End	Mail Body Size	Mail Body Html Size	Mail Attachment Count	Mail Attachment Size	Mail Send Error	Last Updated On	App Id	Worksp
29019155138210412958	D54FEED82379135FE053E73F72642FFC@apex.oracle.com	t.henwix@googlemail.com	noreply@oracle.com		Hello Report			1/11/2022	1/11/2022	1/11/2022	70	4241	0	0		1/11/2022	83036	125322196128847
28964117103803351531	D54D5F68A5C32738E053ED3F7264B113@apex.oracle.com	t.henwix@googlemail.com	noreply@oracle.com		Hello Report			1/11/2022	1/11/2022	1/11/2022	77	4248	0	0		1/11/2022	83036	125322196128847
28963371830819312791	D54D47DAD267D0E2E053ED3F7264919F@apex.oracle.com	t.henwix@googlemail.com	noreply@oracle.com		Hello Report			1/11/2022	1/11/2022	1/11/2022	77	4248	0	0		1/11/2022	83036	125322196128847
28962166279317300842	D54D408BA7AEA595E053ED3F7264122D@apex.oracle.com	t.henwix@googlemail.com	noreply@oracle.com		Hello Report			1/11/2022	1/11/2022	1/11/2022	77	4248	0	0		1/11/2022	83036	125322196128847
27407712693033774529	D4D79D0AE0EC9CD7E053ED3F7264CC6D@apex.oracle.com	t.henwix@googlemail.com	noreply@oracle.com		Hello Report			1/5/2022	1/5/2022	1/5/2022	77	4248	0	0		1/5/2022	83036	125322196128847

1 - 5

Q  Go Actions

Mail To	Mail From	Mail Subj	Mail Message Created	Mail Send Error	Last Updated On	App Id	Workspace Id	Workspace Name	Id	Mail Body	Mail Send Count	Last Updated By	Mail Message Created By
abc	noreply@oracle.com	Hello World	1/13/2022	ORA-29279: SMTP permanent error: 550 5.1.1 unknown or illegal alias: abc	1/13/2022	83036	12532219612884793896	HERWIX	29615467893581599702	Hello Timo, this is an email sent from Oracle APEX. Best Regards	2	DEMO	DEMO

# Including a link in an email to an APEX App

# Including a link in an email to an APEX App

```
1 DECLARE
2   l_body_html clob;
3   l_body      clob;
4 BEGIN
5   l_body := 'Hello ' || :P1_CONTACT || ', ' || UTL_TCP.crlf;
6   l_body := l_body || 'this is an email sent from Oracle APEX.' || UTL_TCP.crlf;
7   l_body := l_body || 'Please click the following link to view your email log.' || UTL_TCP.crlf;
8   l_body := l_body || apex_mail.get_instance_url || APEX_UTIL.PREPARE_URL('f?p=&APP_ID.:1:&APP_SESSION.') || UTL_TCP.crlf;
9   l_body := l_body || 'Best Regards';
10
11  l_body_html := '<p>Hello ' || :P1_CONTACT || ', </p>'
12               || 'this is an email sent from <strong>Oracle APEX</strong>.<br />'
13               || 'Please click the button to view your email log.<br />'
14               || '<table border="0" cellspacing="0" cellpadding="0">'
15               || '   <tr>'
16               || '     <td style="padding: 12px 18px 12px 18px; border-radius:10px; background-color: #056ac8;" align="center">'
17               || '       <a href="' || apex_mail.get_instance_url || apex_page.get_url(p_page => 1) || '" target="_blank">Click here</a>'
18               || '     </td>'
19               || '   </tr>'
20               || '</table>'
21               || 'Best Regards';
22
23  apex_mail.send(
24    p_to      => :P1_TO,
25    p_from    => :APP_EMAIL,
26    p_body    => l_body,
27    p_body_html => l_body_html,
28    p_subj    => 'Including a link in an email to an APEX App'
29  );
30
31  apex_mail.push_queue;
32 END;
```

# Including a link in an email to an APEX App

Hello Timo Herwix,

this is an email sent from **Oracle APEX**.

Please click the button to view your email log.

[Click here](#)

Best Regards

# Generated E-Mails with AI

# Generated E-Mails with AI

- Supports flexible use of user input or dynamically generated content from page or application items
- Combines a System Prompt with values from items
- Generates context-related output based on the provided input
- Displays the result in another item
- Enables dynamic, personalized interactions within your application

The image shows a configuration panel for an AI-generated email feature. It is divided into several sections:

- Identification:** Includes a 'Name' field and an 'Action' dropdown set to 'Generate Text With AI'.
- Generative AI:** Includes a 'Configuration' dropdown set to '- Select -', a 'Service' dropdown set to 'OpenAI', and a 'System Prompt' text area. The prompt reads: "Based on the information provided, compose a formal, professional response email incorporating the following details: 1. Sender's identity and role (if relevant). 2. Core message or purpose of the original communication. 3. Key points or inquiries to address in the reply. 4. Desired tone (e.g., courteous, assertive, conciliatory) 5. Additional context or relevant information to seamlessly integrate into the response. 6. The output text should be rendered as HTML. Please also note formatting, line breaks, etc. Please use the following signature Timo Herwix".
- Input Value:** Includes a 'Type' dropdown set to 'Item' and an 'Item' dropdown set to 'P120\_EMAIL'.
- Use Response:** Includes a 'Type' dropdown set to 'Item' and an 'Item' dropdown set to 'P120\_REPLY'.
- Suppress Change Event:** A toggle switch that is currently turned off.

# Generated E-Mails with AI

Email

Subject: Urgent: Password Recovery Assistance Required

Dear Our Company,

I hope this message finds you well.

I am writing to request your urgent assistance in recovering my password for my account with your services.

Unfortunately, I am currently unable to access my account and have been unsuccessful in using the standard password recovery options provided.

Here are my account details for your reference:

Account Holder Name: John Doe

Username/Email Associated with Account: john.doe@example.com

Could you please assist me in resetting my password at your earliest convenience?

If there are any additional steps or verification processes required, please let me know so that I can promptly provide the necessary information.

Thank you for your immediate attention to this matter. I appreciate your help and look forward to resolving this issue as soon as possible.

Best regards,

Edward Logan

Edward.Logan@example.com

(123) 456-7890

Output

Subject: Re: Urgent: Password Recovery Assistance Required

Dear Mr. Logan,

Thank you for reaching out regarding the password recovery issue. For data protection and security reasons, we are unfortunately unable to reset or modify account credentials directly via email, and we also cannot process requests made on behalf of another person's account (in this case, the account of John Doe).

To proceed securely, please ask the account holder (John Doe) to:

1. Use the "Forgot password" or "Reset password" function on our login page, ensuring that the correct username/email address (john.doe@example.com) is entered.
2. Check the inbox (and spam/junk folder) of the associated email account for the password reset link. This link will guide the account holder through creating a new password.
3. If no reset email is received within a few minutes, please ensure that:
  - The email address on record is active and accessible.
  - Our domain is added to the safe senders list, if applicable.

If John Doe is still unable to reset the password after following these steps, the account holder should contact us directly from the email address associated with the account (john.doe@example.com) so we can guide them through our verification and recovery process in line with our security policies.

Please let us know if you have any further questions regarding the process.

Kind regards,

Timo Herwix

Senior Consultant

T +49 2102 30 961-0

M +49 176 20185455

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



# 1

Basic knowledge  
you should know

# 2

Advanced knowledge  
you can know

# 3

Ready to take off

# Advanced Email Techniques

# Template Directives

Add conditional logic to email templates to dynamically display content in your emails.

- **Clean Code**

Keep your SQL as simple as possible

- **Security**

No RAW syntax required, reducing the risk of cross-site scripting

- **Performance**

Reduce the size of SQL result sets

- **Maintenance**

Reduce the number of email templates

```
1 {if DESCRIPTION/}
2   #DESCRIPTION#
3 {else/}
4   No description.
5 {endif/}
```

```
1 {case JOB/}
2   {when SALESMAN/}
3     #SAL# (#COMM#)
4   {when PRESIDENT/}
5     --
6   {otherwise/}
7     #SAL#
8 {endcase/}
```

```
1 <ul class="tags">
2   {loop ", " OFFERS/}
3     <li class="offer-item">#APEX$ITEM#</li>
4   {endloop/}
5 </ul>
```

# Best Practices

Consider the limitations of different email clients

Prefer `<br>` or `<table>` tags instead of `<p>` or `<span>`

Use hexadecimal color values (e.g. `#000000`)

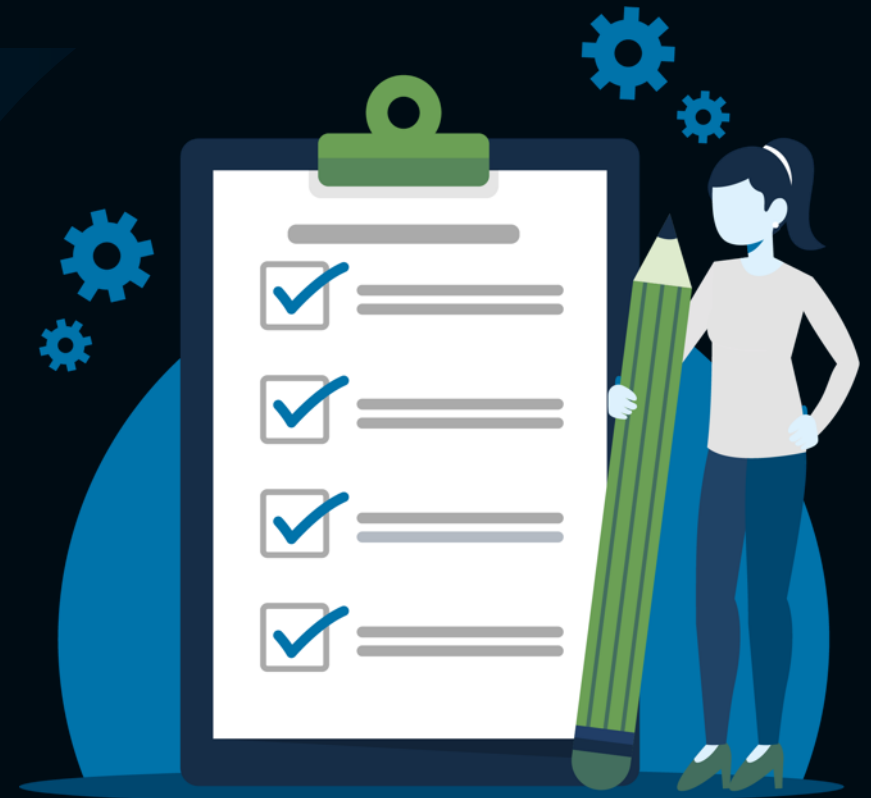
Use inline CSS

Avoid using a single HTML template for all emails

Use loops for mass distribution

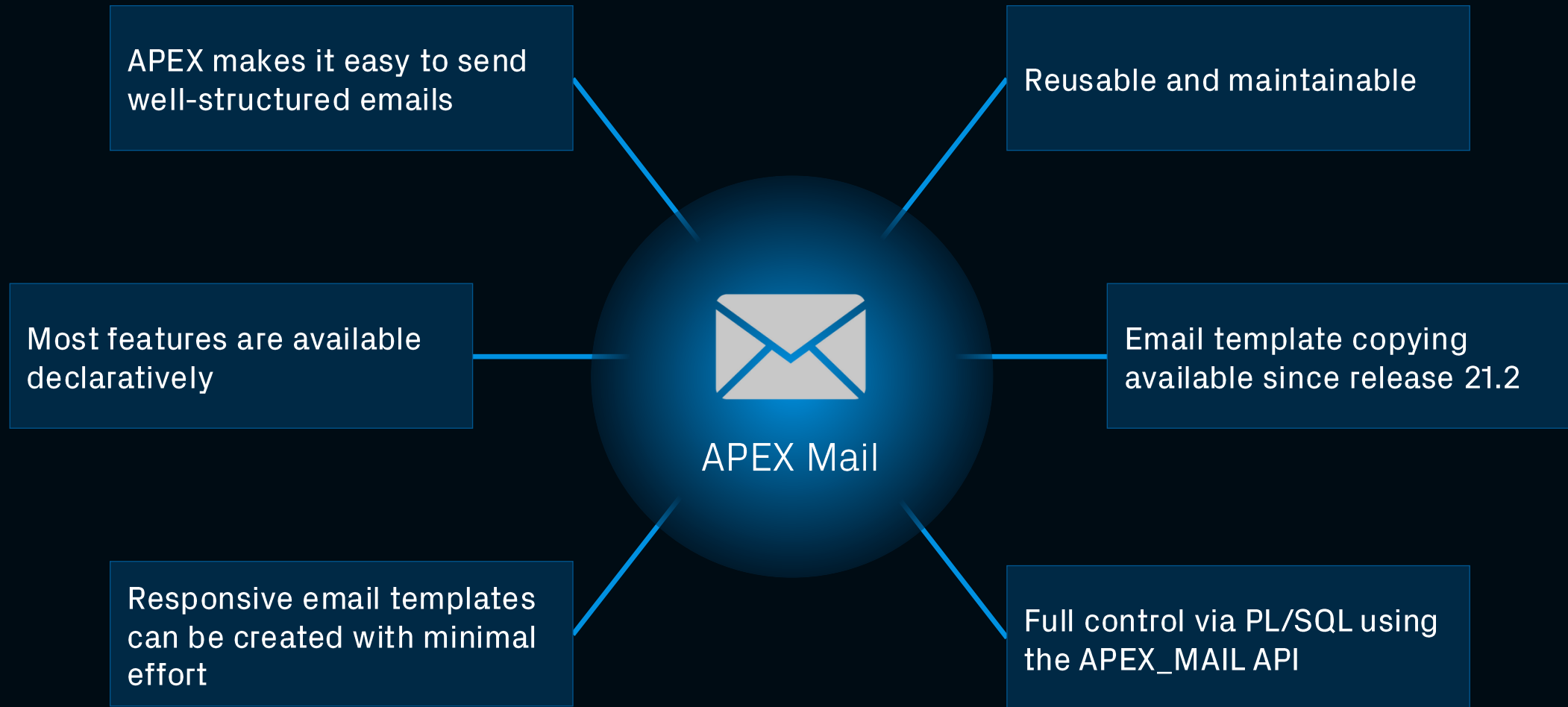
Use SQL techniques to generate dynamic content

Check the expected email size when sending attachments or images



Wrap-Up

# Wrap-Up

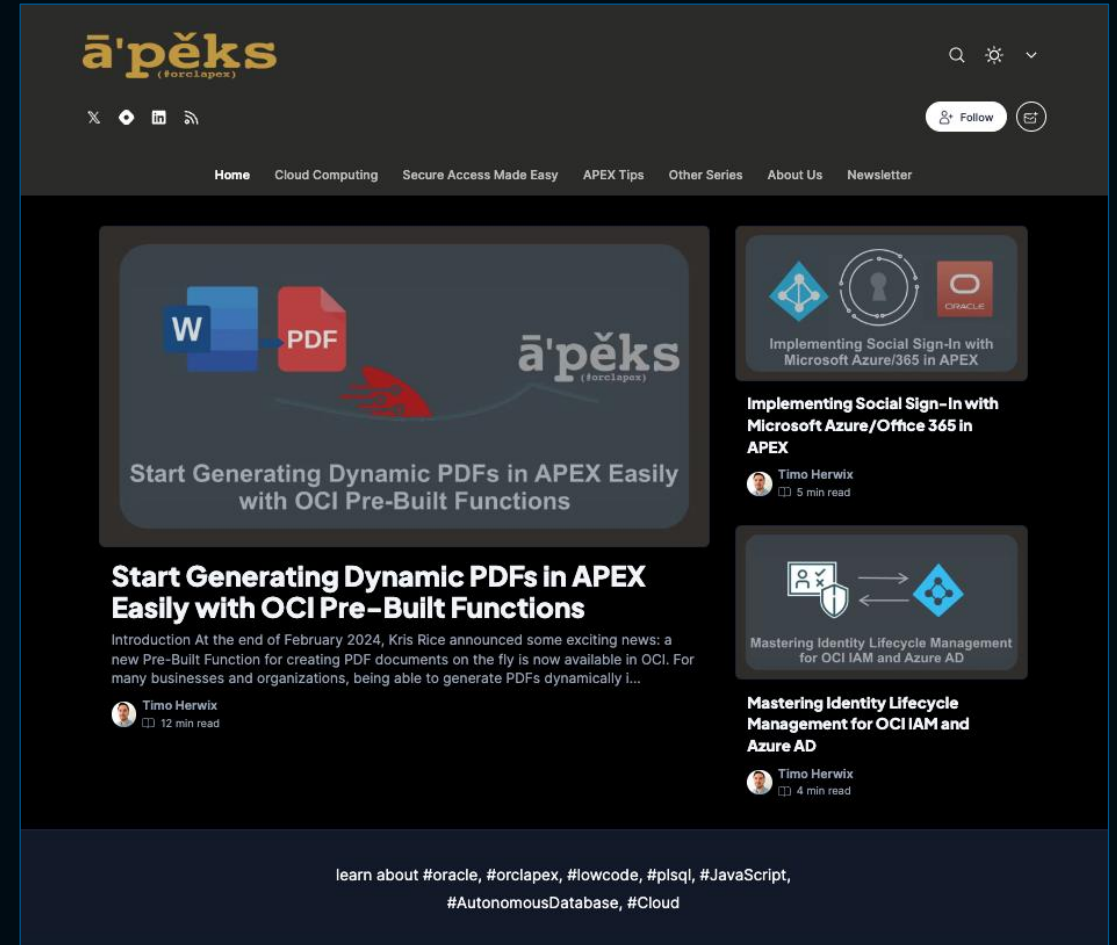


# Blog



Scan me!

tm-apex.hashnode.dev



The screenshot shows the Hashnode profile page for 'ā'pěks' (@orclapex). The profile includes a navigation menu with links for Home, Cloud Computing, Secure Access Made Easy, APEX Tips, Other Series, About Us, and Newsletter. The main content area features three article cards:

- Start Generating Dynamic PDFs in APEX Easily with OCI Pre-Built Functions** by Timo Herwix (12 min read). The article introduction states: "Introduction At the end of February 2024, Kris Rice announced some exciting news: a new Pre-Built Function for creating PDF documents on the fly is now available in OCI. For many businesses and organizations, being able to generate PDFs dynamically i..."
- Implementing Social Sign-In with Microsoft Azure/Office 365 in APEX** by Timo Herwix (5 min read).
- Mastering Identity Lifecycle Management for OCI IAM and Azure AD** by Timo Herwix (4 min read).

At the bottom of the page, there is a footer with the text: "learn about #oracle, #orclapex, #lowcode, #plsql, #JavaScript, #AutonomousDatabase, #Cloud".

# Are you interested?



**Timo Herwix**  
Senior Consultant

Telefon: +49 2102 30 961-0

Mobil: +49 176 20185455

Mail: [timo.herwix@hyand.com](mailto:timo.herwix@hyand.com)



Timo Herwix



@Therwix



[tm-apex.hashnode.dev](https://tm-apex.hashnode.dev)



**Please fill in your  
evaluations**