

# Step-by-Step Guide: Adding a New Document Template

## Example: Executor's Bond Agreement

### Step 1: Create the HTML Template

Location: templates/document\_templates/executors\_bond\_agreement.html

Recommended: Check other templates in this folder for structure and variables.

Example:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Executor's Bond Agreement</title>
  <style>
    @page { size: A4; margin: 2.5cm; }
    body { font-family: 'Times New Roman', Times, serif; font-size: 12pt; }
  </style>
</head>
<body>
  <div class="date-line">{{ today }}</div>
  <div class="solicitor-address">
    {% if solicitor.name %}<div>{{ solicitor.name }}</div>{% endif %}
    {% if solicitor.address %}
      <div>{{ solicitor.address.line1 }}</div>
      {% if solicitor.address.line2 %}<div>{{ solicitor.address.line2 }}</div>{% endif %}
      <div>{{ solicitor.address.town_city }}</div>
      {% if solicitor.address.county %}<div>Co. {{ solicitor.address.county }}</div>{% endif %}
    {% endif %}
  </div>
  <div class="subject-line"><strong><u>EXECUTOR'S BOND AGREEMENT</u></strong></div>
  <div class="subject-line"><strong><u>Estate of {{ deceased.name_with_deceased }}</u></strong></div>
  <div class="subject-line"><strong><u>Application Number: {{ loan_application_number }}</u></strong></div>
  <div class="main-content">
    I/We, the undersigned executor(s) of the estate of {{ deceased.name_with_deceased }}, hereby u
  </div>
  <div class="signature-section">
    <p><strong>Executor(s):</strong></p>
    {% for beneficiary in beneficiaries %}
      <div style="margin-bottom: 40px;">
        <p>Name: {{ beneficiary.full_name }}</p>
        <p>Signature: _____</p>
        <p>Date: _____</p>
      </div>
    {% endfor %}
    <div style="margin-top: 60px;">
      <p><strong>Witnessed by:</strong></p>
      <p>Witness Signature: _____</p>
      <p>Witness Name: _____</p>
      <p>Witness Address: _____</p>
    </div>
  </div>
</body>
</html>
```

# Step-by-Step Guide: Adding a New Document Template

```
</div>
</div>
</body>
</html>
```

## Step 2: Update DocumentTemplateService (document\_requirements/services.py)

Update or add support for your new document in these methods:

- can\_generate\_template
- generate\_pdf\_response
- get\_filename

Optionally, add a dedicated method (e.g. \_generate\_executors\_bond\_agreement\_pdf).

Sample changes:

```
@classmethod
def can_generate_template(cls, document_type):
    supported_templates = [
        "Beneficiaries Authorisation",
        "Executor's Bond Agreement", # ADD THIS LINE
    ]
    if not document_type.has_template:
        return False
    return any(template_name in document_type.name for template_name in supported_templates)
```

```
@classmethod
def generate_pdf_response(cls, requirement):
    try:
        if not requirement.document_type.has_template:
            return None
        if "Beneficiaries Authorisation" in requirement.document_type.name:
            return cls._generate_beneficiaries_authorisation_pdf(requirement)
        elif "Executor's Bond Agreement" in requirement.document_type.name:
            return cls._generate_executors_bond_agreement_pdf(requirement)
        else:
            return None
    except Exception as e:
        logger.error(f"Error generating template PDF: {str(e)}")
        return None
```

```
@classmethod
def _generate_executors_bond_agreement_pdf(cls, requirement):
    try:
        context = cls._get_template_context(requirement)
        html_string = render_to_string(
            'document_templates/executors_bond_agreement.html', context)
        result = BytesIO()
        pdf = pisa.CreatePDF(BytesIO(html_string.encode("UTF-8")), dest=result)
        if pdf.err:
            return None
        return result
```

# Step-by-Step Guide: Adding a New Document Template

```
except Exception as e:
    logger.error(f"Error generating Executor's Bond Agreement PDF: {str(e)}")
    return None

@classmethod
def get_filename(cls, requirement):
    application = requirement.application
    if "Beneficiaries Authorisation" in requirement.document_type.name:
        return f"Beneficiaries_Authorisation_App_{application.id}.pdf"
    elif "Executor's Bond Agreement" in requirement.document_type.name:
        return f"Executors_Bond_Agreement_App_{application.id}.pdf"
    return f"{requirement.document_type.name.replace(' ', '_')}_{application.id}.pdf"
```

## Step 3: Create Document Type in Django Admin

In Django Admin:

- Go to Document Requirements > Document Types
- Click 'Add Document type'
- Fill in fields: Name, Description, Order
- Check 'Is active', 'Signature required', 'Has template'
- Set 'Who needs to sign' as appropriate (e.g., Applicant)

Example Template Fields:

```
{
    "{BOND_AMOUNT}": "application.amount",
    "{EXECUTOR_TITLE}": "applicants.0.title"
}
```

## Step 4: Add the Requirement to an Application

- Go to Application Document Requirements in admin.
- Add a new Application document requirement: select the Application and your new document type. Save.

## Step 5: Download & Test the Template

- In admin, check that Template Status shows 'Can Generate'.
- Template Actions should include 'Download PDF Template'.
- Download and open the PDF. Check all dynamic fields and formatting.

## Step 6: Frontend Check (If Applicable)

- Confirm the 'Auto-Generate' badge appears for your requirement.
- Test that the 'Get Template' button downloads your new document.

Sample API response for a template requirement:

```
{
    "template_available": true,
    "template_download_url": "/api/applications/123/document-requirements/456/download-template/",
    "template_filename": "Executors_Bond_Agreement_App_123.pdf"
}
```

## Step 7: Troubleshooting

- If the template fails to generate:

# Step-by-Step Guide: Adding a New Document Template

---

- Check Django logs for errors.
- Verify your template file exists and is named correctly.
- Ensure xhtml2pdf is installed.
- Double-check template syntax (all {% if %}, {% for %}, etc. are closed).
- For download issues:
  - Check browser network tab for HTTP 500 errors.
  - Ensure admin URLs and tokens are correct.
  - For filename errors, update get\_filename and clear cache.

## Checklist (For Each New Document)

- ☐ Create HTML template in templates/document\_templates/
- ☐ Update can\_generate\_template method
- ☐ Add generation method in services if needed
- ☐ Update get\_filename method
- ☐ Register Document Type in admin
- ☐ Test download in admin
- ☐ Test in frontend (if used)
- ☐ Verify filename and document content

Tip: For styling and context, always review similar templates already present in the project.