**API Endpoints**

**Note - For all users passwords except superuser is password@123**

**Compliance user Create Guidelines**

Should be logged in as compliance user

creds – complianceuser01, complianceuser02, password -

POST*/*api*/*content\_management/listCreateGuideline/

{

"tag": "Copyright and intellectual property rights",

"description": "User should not add copyright material in their content"

}

Response -

{

"id": 3,

"tag": "Copyright and intellectual property rights",

"description": "User should not add copyright material in their content"

}

**Guidelines listing -**

Any user can use.

GET /api/content\_management/listCreateGuideline/

Response-

[

{

"id": 1,

"tag": "Guideline for Job Description",

"description": "Do not use discriminatory language in your content"

},

{

"id": 2,

"tag": "Investment products guideline",

"description": "Content should not make any promises about rate of return"

},

{

"id": 3,

"tag": "Copyright and intellectual property rights",

"description": "User should not add copyright material in their content"

}

]

**View details of particular Guideline (Any user)-**

GET /api/content\_management/retrieveUpdateDeleteGuideline/1/

Response-

{

"tag": "Guideline for Job Description",

"description": "Do not use discriminatory language in your content"

}

**Update a guideline (Compliance user only)**

PATCH /api/content\_management/retrieveUpdateDeleteGuideline/1/

Response-

{

"description": "Do not use discriminatory language in your content. EDITED"

}

**Delete a guideline (Compliance user only)**

DELETE /api/content\_management/retrieveUpdateDeleteGuideline/1/

**Content Author** **user Create Content**

Should be logged in as author user

creds – author01, author02, author03

POST /api/content\_management/listCreateContent/

User multipart formdata fields –

title – char

content\_file – file type

is\_submitted (optional) – true or false

Response-

{

"id": 5,

"title": "JD for job",

"content\_file": "http://127.0.0.1:8000/media/author\_content\_files/sample1.pdf",

"status": null,

"content\_guidelines\_actions": [],

"is\_submitted": false

}

**Content listing -**

Any user can use.

GET /api/content\_management/listCreateContent/

Can be filtered on these fields - ['is\_submitted', 'status']. is\_submitted (true/false), status (PASSED/FAILED)

Search\_fields = ['title']

Response-

[

{

"id": 4,

"title": "samplefile01",

"content\_file": "http://127.0.0.1:8000/media/author\_content\_files/sample-1\_rjoKtb9.txt",

"status": "FAILED",

"content\_guidelines\_actions": [

{

"guideline": 1,

"is\_approved": true

},

{

"guideline": 2,

"is\_approved": false

}

],

"is\_submitted": true

},

{

"id": 5,

"title": "JD for job",

"content\_file": "http://127.0.0.1:8000/media/author\_content\_files/sample1.pdf",

"status": null,

"content\_guidelines\_actions": [],

"is\_submitted": false

}

]

**View details of particular content (Any user)-**

GET /api/content\_management/retrieveUpdateDeleteContent/5

Response-

{

"title": "JD for job",

"content\_file": "http://127.0.0.1:8000/media/author\_content\_files/sample1.pdf",

"is\_submitted": false,

"content\_guidelines\_actions": [],

"status": null,

"created\_by": 4

}

**Update a content (****Author only who created the content)**

PATCH /api/content\_management/retrieveUpdateDeleteContent/5

May use multipart if file changes, otherwise may use json

{

"is\_submitted": true

}

Response-

{

"title": "JD for job",

"content\_file": "http://127.0.0.1:8000/media/author\_content\_files/sample1.pdf",

"is\_submitted": true,

"content\_guidelines\_actions": [],

"status": null,

"created\_by": 4

}

**Delete a content (Author only who created the content)**

DELETE /api/content\_management/retrieveUpdateDeleteContent/5

**Mark a content as PASSED/FAILED against all guidelines (Reviewer only)**

**creds – reviewer01, reviewer02**

First the reviewer would hit the list content api to get all contents which arenot reviewed-

/api/content\_management/listCreateContent/?status=NOT\_REVIEWED

Then he may choose to review a particular content as below-

POST /api/content\_management/contentGuidelinesBulkApprovalV2/

{

"content": 5,

"guidelines\_passed": [

1,2

],

"guidelines\_failed": [

3

]

}

Response-

{

"content": 5,

"status": "FAILED",

"guidelines\_approval": [

{

"id": 20,

"guideline": 1,

"is\_approved": true

},

{

"id": 21,

"guideline": 2,

"is\_approved": true

},

{

"id": 22,

"guideline": 3,

"is\_approved": false

}

]

}

If we check the content detail api now-

GET /api/content\_management/retrieveUpdateDeleteContent/5

Response-

{

"title": "JD for job",

"content\_file": "http://127.0.0.1:8000/media/author\_content\_files/sample1.pdf",

"is\_submitted": true,

"content\_guidelines\_actions": [

{

"guideline": 1,

"is\_approved": true

},

{

"guideline": 2,

"is\_approved": true

},

{

"guideline": 3,

"is\_approved": false

}

],

"status": "FAILED",

"created\_by": 4

}

Here status of the content FAILED because all the guidelines could not PASS.

**Using Viewsets**

**Compliance user Create Guidelines**

Should be logged in as compliance user

creds – complianceuser01, complianceuser02, password -

POST/api/content\_management/guidelines/

{

"tag": "Copyright and intellectual property rights",

"description": "User should not add copyright material in their content"

}

Response -

{

"id": 3,

"tag": "Copyright and intellectual property rights",

"description": "User should not add copyright material in their content"

}

**Guidelines listing -**

Any user can use.

GET /api/content\_management/guidelines/

Response-

[

{

"id": 1,

"tag": "Guideline for Job Description",

"description": "Do not use discriminatory language in your content"

},

{

"id": 2,

"tag": "Investment products guideline",

"description": "Content should not make any promises about rate of return"

},

{

"id": 3,

"tag": "Copyright and intellectual property rights",

"description": "User should not add copyright material in their content"

},

{

"id": 4,

"tag": "test guideline 4",

"description": "This is guideline 4"

}

]

**View details of particular Guideline (Any user)-**

GET /api/content\_management/guidelines/1/

Response-

{

"tag": "Guideline for Job Description",

"description": "Do not use discriminatory language in your content"

}

**Update a guideline (Compliance user only)**

PATCH /api/content\_management/guidelines/1/

Payload-

{

"description": "Do not use discriminatory language in your content. EDITED"

}

Response-

{

"description": "Do not use discriminatory language in your content. EDITED"

}

**Delete a guideline (Compliance user only)**

DELETE /api/content\_management/guidelines/1/

**Content Author** **user Create Content**

Should be logged in as author user

creds – author01, author02, author03

POST /api/content\_management/listCreateContent/

User multipart formdata fields –

title – char

content\_file – file type

is\_submitted (optional) – true or false

Response-

{

"id": 6,

"title": "author01 content 3",

"content\_file": "http://127.0.0.1:8000/media/author\_content\_files/sample-3.txt",

"status": "NOT\_REVIEWED",

"content\_guidelines\_actions": [],

"is\_submitted": false

}

**Content listing -**

Any user can use.

GET /api/content\_management/contents/

Can be filtered on these fields - ['is\_submitted', 'status']. is\_submitted (true/false), status (PASSED/FAILED)

Search\_fields = ['title']

Response-

[

{

"id": 4,

"title": "samplefile01",

"content\_file": "http://127.0.0.1:8000/media/author\_content\_files/sample-1\_rjoKtb9.txt",

"status": "FAILED",

"content\_guidelines\_actions": [

{

"guideline": 1,

"is\_approved": true

},

{

"guideline": 2,

"is\_approved": false

}

],

"is\_submitted": true

},

{

"id": 5,

"title": "JD for job",

"content\_file": "http://127.0.0.1:8000/media/author\_content\_files/sample1.pdf",

"status": "FAILED",

"content\_guidelines\_actions": [

{

"guideline": 1,

"is\_approved": true

},

{

"guideline": 2,

"is\_approved": true

},

{

"guideline": 3,

"is\_approved": false

}

],

"is\_submitted": true

},

{

"id": 6,

"title": "author01 content 3",

"content\_file": "http://127.0.0.1:8000/media/author\_content\_files/sample-3.txt",

"status": "NOT\_REVIEWED",

"content\_guidelines\_actions": [],

"is\_submitted": false

}

]

**View details of particular content (Any user)-**

GET /api/content\_management/contents/5/

Response-

{

"title": "JD for job",

"content\_file": "http://127.0.0.1:8000/media/author\_content\_files/sample1.pdf",

"is\_submitted": true,

"content\_guidelines\_actions": [

{

"guideline": 1,

"is\_approved": true

},

{

"guideline": 2,

"is\_approved": true

},

{

"guideline": 3,

"is\_approved": false

}

],

"status": "FAILED",

"created\_by": 4

}

**Update a content (****Author only who created the content)**

PATCH /api/content\_management/contents/6/

May use multipart if file changes, otherwise may use json

{

"is\_submitted": true

}

Response-

{

"title": "author01 content 3",

"content\_file": "http://127.0.0.1:8000/media/author\_content\_files/sample-3.txt",

"is\_submitted": true,

"content\_guidelines\_actions": [],

"status": "NOT\_REVIEWED",

"created\_by": 4

}

After a content is submitted, author can not update it-

{

"non\_field\_errors": [

"Can not update after content has been submitted."

]

}

**Delete a content (Author only who created the content)**

DELETE /api/content\_management/contents/6/

**Mark a content as PASSED/FAILED against all guidelines (Reviewer only)**

**creds – reviewer01, reviewer02**

First the reviewer would hit the list content api to get all contents which arenot reviewed-

/api/content\_management/contents/?status=NOT\_REVIEWED

Then he may choose to review a particular content as below-

POST /api/content\_management/contentGuidelinesBulkApprovalsViewset/

{

"content": 6,

"guidelines\_passed": [

1,2,3,4

],

"guidelines\_failed": [ ]

}

Response-

{

"content": 6,

"status": "PASSED",

"guidelines\_approval": [

{

"id": 23,

"guideline": 1,

"is\_approved": true

},

{

"id": 24,

"guideline": 2,

"is\_approved": true

},

{

"id": 25,

"guideline": 3,

"is\_approved": true

},

{

"id": 26,

"guideline": 4,

"is\_approved": true

}

]

}

If we check the content detail api now-

GET /api/content\_management/contents/6/

Response-

{

"title": "author01 content 3",

"content\_file": "http://127.0.0.1:8000/media/author\_content\_files/sample-3.txt",

"is\_submitted": true,

"content\_guidelines\_actions": [

{

"guideline": 1,

"is\_approved": true

},

{

"guideline": 2,

"is\_approved": true

},

{

"guideline": 3,

"is\_approved": true

},

{

"guideline": 4,

"is\_approved": true

}

],

"status": "PASSED",

"created\_by": 4

}

Here status of the content PASSED because all the guidelines passed..