Doccano问题及解决方案

问题1：原始的应用在进行文本标注分类的时候，json格式时对应的label输出为数字编码格式的数据，但我们的目标是需要label的具体名称。

解决方案：通过对源码的分析，发现/app/api/utils.py文件编写的是数据上传以及下载调用的代码，后续如果需要对数据上传形式进行修改，可以在此文件下

进行修改。为了改变json格式文件的下载方式，需要修改同根下的views.py文件中的TextDownloadAPI类下的get函数以及其调用的在JSONPainter.paint函数，类似JSONPainter.paint\_labels函数，需要接入labels参数进行类似的修改。同时，对user也进行类似的修改，使得下载的数据对应了用户名，而不是其编号。

|  |
| --- |
| get函数的修改 |
| def get(self, request, \*args, \*\*kwargs):  format = request.query\_params.get('q')  project = get\_object\_or\_404(Project, pk=self.kwargs['project\_id'])  documents = project.documents.all()  painter = self.select\_painter(format)  # json1 format prints text labels while json format prints annotations with label ids  # json1 format - "labels": [[0, 15, "PERSON"], ..]  # json format - "annotations": [{"label": 5, "start\_offset": 0, "end\_offset": 2, "user": 1},..]  labels = project.labels.all()  users = project.users.all()  if format == "json1":  data = painter.paint\_labels(documents, labels, users)  else:  data = painter.paint(documents, labels, users)  return Response(data) |

|  |
| --- |
| JSONPainter两个函数的修改 |
| def paint(self, documents, labels, users):  serializer\_labels = LabelSerializer(labels, many=True)  serializer\_users = UserSerializer(users, many=True)  serializer = DocumentSerializer(documents, many=True)  data = []  for d in serializer.data:  d['meta'] = json.loads(d['meta'])  for a in d['annotations']:  label\_obj = [x for x in serializer\_labels.data if x['id'] == a['label']][0]  user\_obj = [x for x in serializer\_users.data if x['id'] == a['user']][0]  a['label'] = label\_obj['text']  a['user'] = user\_obj['username']  a.pop('id')  a.pop('prob')  a.pop('document')  data.append(d)  return data  @staticmethod  def paint\_labels(documents, labels, users):  serializer\_labels = LabelSerializer(labels, many=True)  serializer\_users = UserSerializer(users, many=True)  serializer = DocumentSerializer(documents, many=True)  data = []  for d in serializer.data:  labels = []  for a in d['annotations']:  label\_obj = [x for x in serializer\_labels.data if x['id'] == a['label']][0]  user\_obj = [x for x in serializer\_users.data if x['id'] == a['user']][0]  d['user'] = user\_obj['username']  label\_text = label\_obj['text']  label\_start = a['start\_offset']  label\_end = a['end\_offset']  labels.append([label\_start, label\_end, label\_text])  d.pop('annotations')  d['labels'] = labels  d['meta'] = json.loads(d['meta'])  data.append(d)  return data |

类似地，如果需要对csv格式的下载函数CSVPainter.paint函数进行修改。

|  |
| --- |
| CSVPainter.paint函数的修改 |
| def paint(self, documents, labels, users):  data = super().paint(documents, labels, users)  res = []  for d in data:  annotations = d.pop('annotations')  for a in annotations:  res.append({\*\*d, \*\*a})  return res |

问题2：对所有项目下的全部数据进行批量删除会出错。

在app/settings.py中设置DATA\_UPLOAD\_MAX\_NUMBER\_FIELDS的值，其默认值为1000。

问题3：对标注页面进行跳转功能的设置。

问题定位：/app/server/static/components/annotation.pug line153。该行实现的是在标注数据时标注框中下部分显示[当前页码/总页码]的格式。

问题4：页码跳转功能有bug：在跳转500页等特殊页面时会出现非法页码的提示（future）。

问题5：实体标注时使用快捷键出现卡顿导致标注不能及时打上（紧急）。

问题6：分组管理员权限未设置，即在普通用户和超级管理员中间设置一层中间管理员，该中间管理员只能管理他们组内的项目创建以及项目分配等（future）。