6 Important Factors when Choosing a PDF Library

ADAM PEZ

Overview

You plan to embed PDF functionality into an application. But before you dive into the project, you must decide: do you go with a more expensive commercial PDF SDK — or a lower-cost alternative such as an open-source library or an open-source wrapper?

There are non-trivial costs to switching later. Developers have to re-learn the new library, re-adjust the backend, customize the UI to match what users are accustomed to, as well as migrate documents, form data, annotations, and more.

According to market research conducted by Stax Inc., the average Net Promoter Score (NPS) for the top five PDF SDK vendors is 35%. And 70% of customers express interest in switching despite the high costs.

This dissatisfaction implies that picking the right PDF SDK is a lot harder than it seems. And to help you avoid the same mistakes as past implementations, we've written this article.

(We also recently surveyed 57 unique organizations that switched from PDF.js to a commercial SDK. Read our comprehensive guide to PDF.js to learn more.)

Restrictions on Features and Platforms

A first mistake organizations make when selecting a PDF library for the first time is to assume fixed feature requirements. But these are likely to evolve.

Users start to ask for more functionality as they grow dependent upon a PDF SDK. An organization will then have to consider saying no to user feature requests; building time-intensive and challenging customizations on top; or integrating additional libraries and thus adding more complexity, maintenance overhead, and risk.

Additionally, a library may work fine initially on the main platforms preferred by your users. But later if you wish to expand, the library does not support the platforms you want — or the APIs are inconsistent, with different classes and methods across platforms making it so your engineers have to start from scratch.

To avoid this hidden cost, go with an SDK with a broad feature-set across multiple platforms, providing you the flexibility to grow down the road.

> *Maybe big companies can absorb the costs of maintaining three-to-four different relationships with different vendors, each with a different code base, different roadmaps, and different problems. I'm not saying it isn't possible.*
>
> — Kalsefer Co-Founder and CEO, Avshi Segev

Unanticipated Difficulty Adding Features

Another mistake is where organizations select a basic library to save money with the assumption that they can build anything needed on top.

But building in an unfamiliar domain can easily lead to unknown challenges, high expenses, and reduced speed to market. And PDF is unusually complex as high-profile teams attest — including those of Slack, Dropbox, and Linkedin.

Your devs are not necessarily PDF experts, and attempting challenging PDF features in-house involves a steep learning curve not subject to economies of scale. Throwing more devs into the equation will not shrink the ramp-up time for the first developer.

Additionally, custom features will have to be supported and maintained long-term, creating an additional ongoing opportunity cost: committed resources will be less-available to work on other projects.

> *PDFs are an incredibly complex file format; this is especially so given that a PDF can be generated a hundred different ways, all of which a renderer needs to handle gracefully.*
>
> — Developer, LinkedIn

> *PDFs are complex documents — structured into different layers of information, data, and objects, and containing different languages, images, and graphics.*
>
> — Developer, Slack

> *PDF is an incredibly complex file format — the specification is more than a thousand pages long, not including the extensions and supplements.*
>
> — Developer, Dropbox

Organizations that we've spoken to have found the most challenging features are those that require engaging PDF at a low level, where objects are defined in PDF byte code — with unique byte offsets for different objects, making it difficult for devs unfamiliar with PDF's inner workings to parse and manage these objects correctly.

Challenging PDF functionality includes

- Managing PDF annotations from multiple users (e.g., synchronization and versioning
- PDF generation (creating PDFs from scratch or from other documents)
- Page manipulation (add, merge, or remove)
- Layers (via Optional Content Groups)
- Color management features (e.g., ink-color separations, overprint, etc.)

While it is certainly possible to build the above in-house, PDF features can consume a shocking amount of time. And you eventually may have to decide whether to continue — or whether to bite the bullet and abandon months or years of work for an alternative that can meet your requirements cost-effectively.

To avoid this type of hidden cost, you will want to carefully consider the capacity of your existing development team should you decide to build, maintain, and support custom PDF features in-house as these features often prove time-intensive and challenging.

> *…you shouldn't build anything that's available off the shelf because it's not a source of competitive advantage if everybody else can avail themselves of it. The only scenario where you should build is if it's your core technology -- the core source of your competitive differentiation and competitive advantage.*
>
> — Mark Holst-Knudsen, President ThomasNet @ MIT's 2014 CIO Symposium

A lower-quality library also encounters performance and memory issues, such as large documents with frustratingly long wait times for your users as well as complex documents that crash the viewer. This is often due to the absence of features such as PDF tiling, parallelization, and linearization that a more mature PDF SDK will incorporate.

Some solutions (e.g., image servers) perform excellently when tested on a small number of documents and users but then inflict unexpected hidden costs when scaled up. When hundreds or thousands of users later view, mark up, comment on, and otherwise interact with (i.e.,scroll, pan, and zoom) documents, server resource and network data usage explodes. To maintain your desired UX, you have to pay higher fees or invest in more servers.

The following types of documents have much more demanding rendering requirements:

- CAD-based PDFs such as construction and engineering drawings with very large and complex designs.
- Reports, textbooks, and marketing material using advanced PDF graphics such as shadings, gradients, soft masks, and patterns.
- Geospatial maps with OCG layers that are switched off by default.
- Pre-press documents which require an SDK with advanced color management features to print colors accurately.
- High-speed accurate rendering (especially on native mobile apps and mobile browsers).
- Context extraction of tables, text, etc. with document structure (e.g., text read order or table arrangement) in tact.

To prevent crashes, slowness, and rendering issues from disrupting your UX, test functionality with the types of documents your users will work with. Also test a server-based solution at the anticipated load and usage.

Poor UX: Slow Performance, Crashing, and Inaccurate Rendering

Another source of hidden costs can be a poor user experience, especially as users start to upload more massive and complex documents that crash or freeze a lower-quality viewer. Construction Computer Software encountered these issues with a free PDF viewer add-on to its flagship estimation software.

As is often the case with a lower-quality library, PDFs render incorrectly. You then have to wait on the vendor to respond. But a reseller or a smaller company with many remote developers may have difficulty providing the same turn-around time and specialized support and service as a commercial SDK. If they did not build the rendering engine themselves, they may not be able to fix the issue — or fixes may take a long time — because they have difficulty finding in the code where the problem originated. If you go with open-source, you may have to fix bugs yourself.

> *If you're looking for a PDF reader for the first time, you better make sure it can read 100% of your PDF files. Because if your client-base starts relying on that PDF reader, exactly what happened to us, they still want the absolute best quality."*
>
> — Tony Cornwall, Construction Computer Software

Low Adoption on a Complex UI

In 2018, AEC-software company PlanGrid partnered with FMI to survey nearly 600 construction leaders from around the world to discern why construction and engineering software succeed or fail. The findings report "Construction Disconnected" identified a complex UI and inadequate user training as two of the top five reasons for why technology fails.

Being able to slim down the interface and tailor feature-sets to specific user groups is proven to significantly cut down training costs and improve user adoption. (See our OEC Graphics success story to learn more.)

However, a closed-source UI will limit you in what you can customize, and you may not be able to fully fix the UX. (And by the time you've discovered this, it may be too late.) A closed-source UI will make it difficult to evaluate how deeply you can customize, optimize, and add new tools or annotation types to the UI. Therefore, your team may build out a proof of concept and make their plans for future expansion — only to have to scale back their ambitions or wait on the vendor to adjust the API. A black box UI will prove especially problematic if your UI team is very strict or if you have unique UI requirements (e.g., accessibility compliance requirements such as ADA/508).

To avoid this hidden cost, choose a vendor with an open-source UI or make certain your proof of concept won't need to change.

Security Issues

When writing PDF features from scratch, developers may be tempted to take shortcuts to save time. But these shortcuts cause the solution to become obsolete quickly as devs run into the exact security issues a more mature toolkit makes a lot easier to solve.

One recent instance our solution engineers have noted is where developers use JavaScript-based submit buttons on forms rather than uploading and parsing data out of forms — which opens up the system to phishing and middle-man attacks. Someone could easily edit the button to have it send personal information to another server, and then maliciously re-circulate the form within your organization or send it to end users.

Vendor Lock-in

Lastly, consider how your data and documents will be stored. For example, annotations stored in a proprietary format, such as Brava! annotations and some versions of JSON, will not be accessible to users who want to view their annotations with other tools such as Adobe Acrobat. Moreover, it will be challenging to migrate these annotations later if you wish to switch solutions.

A vendor who manages annotations in the ISO standard for annotations interchange, XFDF, for example, will eliminate this hidden cost.

The Bottom Line

The best way to avoid hidden costs associated with the wrong PDF library is to perform due diligence during your evaluation. To assist you in this process, we've written a blog with several considerations you can add to your PDF SDK evaluation checklist.

We hope this article was helpful! If you have any questions, don't hesitate to contact us.

常住人口登记卡

姓名	刘宏平	户主或与户主关系	子		
曾用名		性别	男		
出生地	江西省抚州市资溪县	民族	汉族		
籍贯	江西省抚州市	出生日期	1997年11月10日		
本市(县)其他住址		宗教信仰	无宗教信仰		
公民身份证件编号	362529199711100054	身高	163厘米	型	B型
文化程度	初中	婚姻状况	未婚	兵役状况	未服兵役
服务处所		职业	学生		
何时由何地迁来本市（县）	2011年12月23日因招工家属随迁由江西省抚州市资溪县鹤城建设中路33号1栋105迁来本市(县)				
何时由何地迁来本址	2011年12月23日因招工家属随迁由江西省抚州市资溪县鹤城建设中路33号1栋105迁来本址				

承办人签章： 叶子芳

登记日期： 20年 1月 2日

深圳市公安局 [illegible]派出所 户口专用

个人信息

姓名：刘宏平　　性别：男　　工作经验：2 年　　学历：本科　　薪资：面议

手机：136-5238-2944(微信同号)　　邮箱：liuhongping2103@163.com

专业技能

1、 熟练掌握 JavaScript、CSS、HTML、ES6。

2、 熟练掌握移动端，PC 端布局，响应式布局，弹性盒布局，媒体查询。

3、 熟练掌握 Vue 框架及相关生态，有基于 Vue2.0 生态的项目架构经验。

4、 熟练掌握 React 框架及相关生态，有基于 React 生态的项目架构经验。

5、 熟悉常用设计模式，面对对象编程，并了解基本数据结构和常用算法。

6、 熟练掌握 ECharts，V-charts，ant-v，Canvas 等可视化插件，有百度地图 GL 使用经验。

7、 了解 Node.js ，C ，PHP 等后端语言，拥有搭建简易服务器的能力。

8、 拥有基本使用 Webpack，Vite，Gulp，Git 等前端架构工具的能力。

9、 了解 WebSocket，WebGl，MySQL，MongoDB，金融及相关概念。

10、 对网络安全有基本认知，在项目开发中会基于此进行深入考虑。

11、 有基于 JQuery 的项目开发经验，并了解 Svelte，Solid 等框架。

12、 熟练掌握可复用插件二次封装以及公共组件封装的能力。

13、 熟练掌握 Ajax，Axios，Fetch，XMLHttpRequest 技术。

14、 了解 web3.js，Solidity，Dapp 开发及区块链相关概念。

15、 熟练掌握 Element ，Ant-Design ，Vant 等 UI 框架。

16、 熟悉 sass 及 css-in-js 解决方案 styled-components。

17、 深刻理解前后端架构分离及模块化开发。

18、 掌握常规前端性能优化方案。

工作经历

2020.7-2022.3	深圳升曦网络科技有限公司	前端开发工程师
2019.5-2020.3	太原多讯网络科技有限公司	前端开发工程师

项目经验

项目：客资运营平台（PC）

技术栈：React + Bigfish + Dva + Redux + Redux-saga + React-Router-Dom + React-Router-Config + TypeScript + Less + Ant-Design + Moment + Mock + Request + I18n + Lodash

平台内部客资账单管理平台，在权限管理公共组件上，在项目内负责架构维护，webpack 配置及插件开发，公共组件开发，实现业务相关功能，并通过自定义 hook 和基于设计模式进行公共组件封装降低工作成本。

项目：企业内部后台管理系统（PC）

技术栈：React + Redux + ES6+ React-Router-Dom + Redux-thunk + prop-types + Redux-saga + Sass + Ant-Design + AntV + IScoll + Mock+ Immutable + Lodash + WangEditor

通过 AntV 引入数据可视化管理，在全局封装自定义 hook 实现金额录入二次校验，在高阶组件和 ref 转发基础上实现权限分组，并在 iscoll 基础上二次封装全局公共组件。通过 Canvas 封装实现页面水印功能，引入富文本编辑器实现技术文档交流区，并通过 WebSocket 实现实时评论及通知。

项目: 黄商超市企业平台(APP)

技术栈：Vue2.0 + VueX + Vue3.0 + Vue-Router + Ts + Sass + Vant UI + Echarts + Better-Scroll + Jspdf + Exif + Fastclick + Js-Bridge + Swiper

超市管理综合服务平台项目，通过联动 PC 端实现二维码登录，内嵌服务端渲染实现快速通知管理，通过媒体查询实现页面适配。并在全局引入错误边界实现生成错误日志通知后台。

项目: 黄商超市企业平台(PC)

技术栈：Vue2.0 + VueX+ Vue-Router + Ts + Scss + Element UI + Echarts + IScroll + Cropper

超市管理综合服务平台 pc 端项目，通过权限管理实现对角色员工及商品进行信息化管理，通过数据可视化实现对商品库存管理，订单管理，审批链管理，员工日常考勤管理等模块实现可视化管理。

教育经历

2016.9~2020.5	中北大学	飞行器制造工程	本科

户　别	家庭户	户主姓名	刘毅	
户　号	641128987	住　址	广东省深圳市宝安区龙华龙发路228号中环花园日明阁708	

广东省公安厅
户口专用

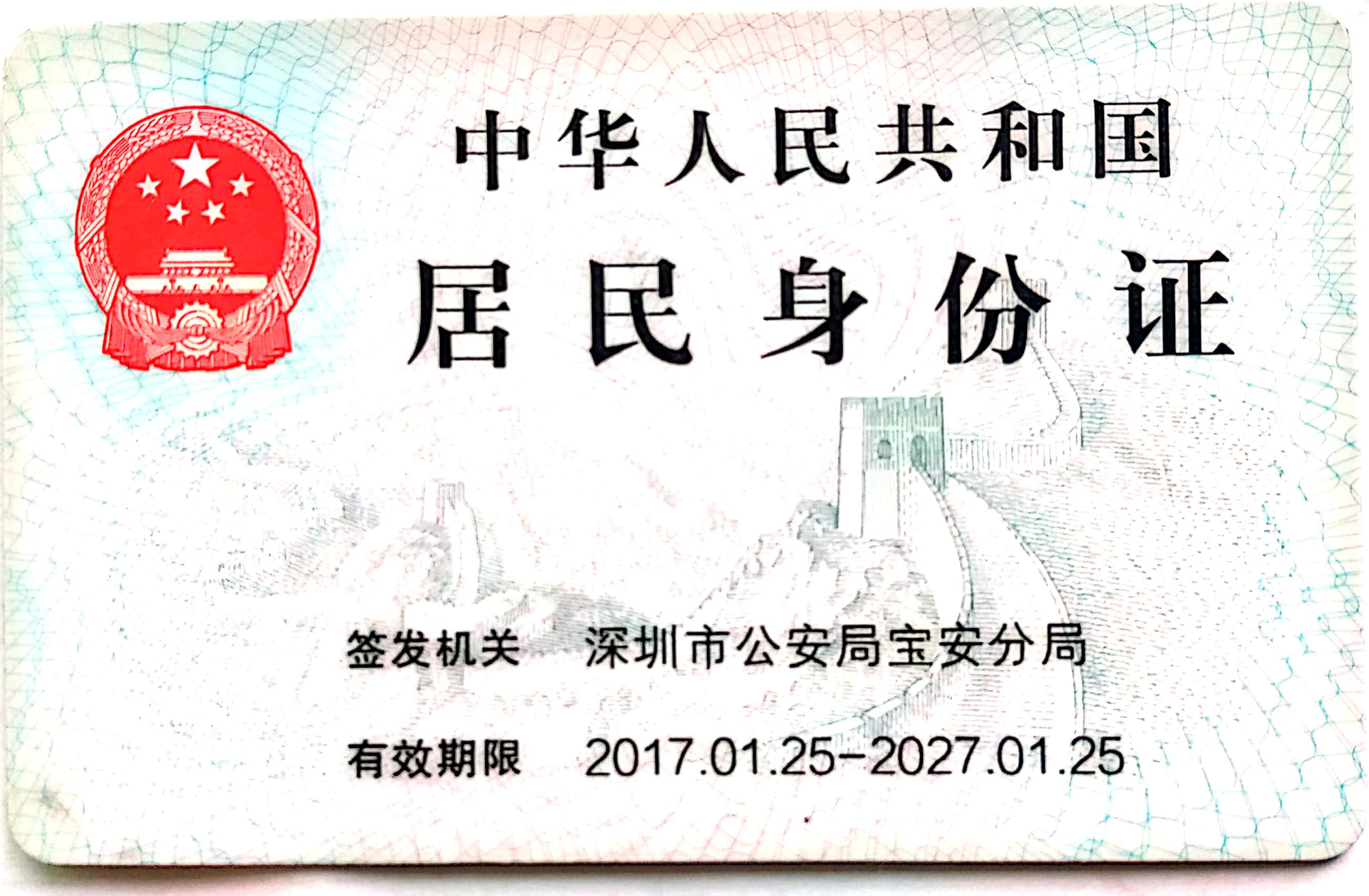
省级公安机关
户口专用章

深圳市公安局
户口专用
宝城派出所

户口登记机关
户口专用章

承办人签章：叶子芳

2011 年 12 月 23 日　签发



中华人民共和国

居民身份证

签发机关 深圳市公安局宝安分局

有效期限 2017.01.25-2027.01.25

2020年全国硕士研究生招生考试网上报名信息

报考点	1405(中北大学)	考生报名号	140598545
考生姓名	刘宏平	考生姓名拼音	liuhongping
证件类型	01(居民身份证)	证件号码	362529199711100054
出生日期	1997-11-10	民族	01(汉族)
性别	1(男)	婚否	1(未婚)
现役军人	0(非现役军人)	政治面貌	03(中国共产主义青年团团员)
籍贯所在地	361028(江西省抚州市资溪县)		
户口所在地	440309(广东省深圳市龙华区)		
户口所在地详细地址	龙华街道中环花园日明阁二单元708		
出生地	361028(江西省抚州市资溪县)		
现在学习或工作单位	中北大学		
考生作弊情况	无		
通讯地址	山西省太原市尖草坪区中北大学		
邮政编码	030051	联系电话	0
移动电话	18406582875	电子信箱	627166963@qq.com
考生来源	5(普通全日制应届本科毕业生)	注册学号	1601024246
符合报考条件的最后学历毕业学校	10110(中北大学)		
符合报考条件的最后学历毕业专业	081503(飞行器制造工程)		
取得符合报考条件的最后学历的学习形式	1(普通全日制)	符合报考条件的最后学历	2(本科毕业)
符合报考条件的最后学历证书编号		符合报考条件的最后毕业日期	2020-07-10
最后学位	4(无)	最后学位证书编号	
报考单位	10055(南开大学)		
报考专业	025100((专业学位)金融)		
考试方式	21(全国统考)	专项计划	0(无)
报考类别	11(非定向就业)		
报考院系所	130(金融学院)		
研究方向	00(不区分研究方向)	学习方式	全日制
政治理论	101(思想政治理论)	外国语	204(英语二)
业务课一	396(经济类联考综合能力)		
业务课二	431(金融学综合)		

何时何地何原因受过何种奖励或处分	无		
家庭主要成员			
姓名	与本人关系	在何单位/任何职务	联系电话
刘毅	父子	无	13828764598
学习与工作经历			
起止年月	学习或工作单位		任何职务
2016年9月-2019年9月	中北大学		无
考生档案所在地	140108(山西省太原市尖草坪区)		
定向就业单位所在地			
定向就业单位			
考生档案所在单位	中北大学		
考生档案所在单位地址及邮政编码	山西省太原市尖学院路3号(030051)		
入伍前所在高校信息			
高校名称及入学年月			
专业名称			
学习形式		毕业证书编号	
入伍批准书编号		批准入伍年月	
批准入伍机关名称			
退出现役证编号		批准退役年月	
批准退役机关名称			
备用信息			
备用信息1			
登录用户名	18406582875	交费标志	已交费
首次报名时间	2019-09-24	最后修改时间	

打印时间：2019-09-24 13:28:56 (仅供参考，以现场确认时信息为准)

编号：　2020L11

终止/解除劳动合同证明

姓　名	刘宏平	原工作单位	洛阳北方企业集团有限公司
身份证号码	362529199711100054	原工作岗位	见习
原合同期限	2020.07.13--2023.07.31	离职原因	辞职
本企业 工作年限	2020.07-2020.12	终止/解除日期	2020年12月28日

洛阳北[illegible]

员工签名：刘宏平　　　　日期：2020 年 12 月 28 日