数据保护官沙龙（DPO Salon）公益出品

法国国家数据保护委员会（CNIL）关于APP开发人员GDPR指南（中译文）

译者：洪延青等

2020年8月

CNIL.
COMMISSION NATIONALE INFORMATIQUE & LIBERTÉS

// La CNIL publie un guide
// RGPD pour les développeurs

开发人员GDPR指南

法国国家数据保护委员会（CNIL）发布了针对开发人员的欧盟通用数据保护条例（GDPR）指南

为了让您的网页或应用程序开发符合法律要求，CNIL制定了相关指南，教您如何正确开发开源项目。

本指南发布许可证为GPLv3许可证和2.0开放许可证（与CC-BY 4.0 FR兼容）。因此，您可以自由地对此指南进行补充和修改。

该指南只针对开发人员么?

无论您是独自开发项目，还是与团队一起，是开发团队管理者，还是开发供应商，或仅仅是出于好奇心的尝试，都可以通过本指南了解GDPR条例的主要原则，以及在部署应用程序时，保护用户隐私需注意的事项。

无论您的机构大小，您都可以通过本指南获取操作建议和方法，理解GDPR条例的核心概念。本指南也可为公司内部规划和维护客户关系提供指导。

该指南包含哪些内容?

该指南分为**16个章节**，基本覆盖开发人员项目实施各个阶段（从开发准备到用户访问量统计）的需求。

欧盟通用数据保护条例（GDPR）规定，对自然人权利和自由的保护要求"**采取适当的技术和组织措施以确保满足该法规的要求**"（第78条）。

采纳哪些数据保护措施**与数据处理的场景息息相关**。因此数据处理者（处理个人数据的机构或个人）必须确保被处理数据的安全性。

该指南中的操作方法不涵盖所有的法规要求也不是强制性的，仅为在IT开发过程中遇到的个人数据保护问题提供最优措施。某些场景下，为确保完全遵守该法规，还需要根据数据处理的性质，采取额外的措施。

目录

如何完善该指南?

该指南有两个版本：

- 点击这里，查看CNIL网站上的网页版本。点击这里，在项目仓库的“Releases”页签下，可获取PDF版本和源代码。
- GitHub版本，允许读者补充和修改。

如果您想要补充和修改，请按照以下步骤操作：

- 在Github平台上注册；
- 进入到该指南项目页面；
- 您可以：
  - 点击页签“Issue”查看评论或加入讨论。
  - 使用“Fork”选项进行修改，并点击“Pull Requests”请求将修改内容纳入其中。

您的修改建议在发布之前会被CNIL委员会审查。该指南的网页版本会定期更新。

如何发布该指南?

如果您想自己发布该项目，可以使用Pandoc工具。通过此工具，您可以将文件转换为docx或HTML格式。

点击这里，查看Pandoc工具安装指南。

- **生成.docx文件:**

  pandoc -s --toc --toc-depth=1 -o Guide_GDPR_developpeur.docx [0-9][0-9]*.md

- **生成.html文件:**

  pandoc -s --template="templates/mytemplate.html" -H templates/pandoc.css -o index.html README.md [0-9][0-9]*.md

预备章节：遵照GDPR条例进行开发

无论您是独自开发项目，还是与团队一起，是开发团队管理者，还是第三方开发供应商，都必须确保在项目的整个生命周期内，用户的个人数据以及对这些数据的所有操作得到永久保护。

以下步骤指导您如何开发保护用户隐私的应用程序或网站:

1. **清楚GDPR条例的主要原则。**如果您是团队合作，我们建议您指定一个人员负责监督合规性。如果您的公司有一位数据保护官（DPO），那么他就是理解和履行GDPR条例义务的核心人物。某些情况下，尤其是您的程序或应用大量处理"敏感"数据（请参见示例）或大范围地进行定期和系统追踪时，您必须指定这样一个人。
2. **对系统内的数据和处理方法进行映射和分类。**准确列出您的程序或应用使用的数据处理方法，对保证您的应用程序履行相关法律义务大有裨益。制作数据处理记录表（参考CNIL网站示例），可以让您对所处理的数据有一个全面认知，从而识别并判定相关风险的优先级。因为，个人数据有可能出现在您预想不到的地方，例如服务器日志上，缓存文件，Excel文件中等。所以，在绝大部分情况下，您都需要填写数据处理记录表。
3. **确定行动的优先级。**根据数据处理记录表，在开发上游，就了解清楚遵守GDPR条例需要执行的相关操作，并着重关注收集对象的个人数据可能面临的风险。需要着重关注的是：收集和处理的数据类型和必要性，处理所基于的法律依据，程序或应用的告知信息，您与供应商的合同条款，用户行使权利的方法以及为保护数据处理安全采取的相关措施。
4. **风险管控。**如果您发现数据处理的方式很可能导致相关用户数据泄露，您需要根据情况，采取适当的措施来规避风险。数据保护影响分析（DPIA）可以帮助您管控风险。CNIL委员会为您提供了方法，文档模板和工具，帮助您识别风险；以及一系列实践案例，教您如何处理这些风险。此外，所有可能对数据收集对象的权利和自由产生高风险的处理操作，必须执行数据保护影响分析。CNIL网站列出了各种数据处理类型，以及是否需要执行DPIA的列表。
5. **开展公司内部流程，以确保开发的每个阶段合规。**公司内部流程需要做到：开发项目的各个方面均注意用户隐私保护，对突发的隐私安全事件有所准备（例如，安全漏洞、用户访问和纠正个人数据的请求、个人数据的修改、供应商的变更、数据泄露等）。您可以查看CNIL管理标签（自GDPR条例生效以来，CNIL不再授予该标签）颁发要求，帮助您获得更多灵感，并在企业内部成立必要的部门。
6. **随时将开发的合规情况记录在程序文档中，以证明您遵守了GDPR条例，**也就是说，您需要准确掌握开发各

个阶段执行的操作和程序文档。您还要定期检查和更新这些程序文档，保证文档中所述的数据和处理方法始终与您软件实际执行的保持一致。

CNIL网站上提供了大量合法处理数据的案例，涉及不同的商业领域，供您查阅。

第一章：识别个人数据

如果您想开发合规的应用程序，理解“个人数据”、“处理目的”和“处理方法”等概念至关重要。特别注意，不要混淆GDPR条例中划分非常精确的2个概念，“匿名”和“假名”。

定义

- **个人性质的数据，**（通常叫做“个人”数据）在欧盟通用数据保护条例（GDPR）中被定义为“任何已识别或可识别的自然人（数据主体）相关的信息”。个人数据既包括能够直接识别自然人的数据（例如姓名），也包括间接识别自然人的数据（例如电话号码、车牌号、终端设备标识等。）

- 对这类数据执行的任何操作（收集、存储、传输、修改、散播等）**均受GDPR条例的约束**并且必须遵守条例中的相关规定。对这些数据的处理必须合法，且有明确目的。所收集的个人数据必须密切相关，并且是达到数据处理目的所严格必须的。

个人数据示例

- **当收集的数据涉及自然人时，以下属于个人数据范畴：**
  - 姓名、别名、生日；
  - 照片、录音；
  - 固定或移动电话号码、邮寄地址、电子邮箱；
  - 地址、登录用户名或使用cookie自动登录的用户名；
  - 指纹、静脉、掌纹、视网膜；
  - 车牌号、社会保险号、身份证号；
  - 应用程序使用情况数据、评论等
- **识别一个自然人可以根据：**
  - 单项个人数据（例如姓名）。
  - 多项个人数据的推断（例如根据某个女性住在某地、什么时候出生、参加什么协会、可以确定这个自然人）。
- 一些数据被认为是**敏感数据。**GDPR条例禁止收集或使用这类数据，除非数据主体明确表示同意（主动的，明确的，且最好以书面形式；意愿的表达必须是自由的、具体的和对相关处理知情的）。
- **以下属于敏感数据范畴：**
  - 有关**个人健康状况**的数据；

- 有关**性生活**或**性取向**的数据；
- 揭示**种族**或**民族**的数据；
- **政治观点、宗教**或**哲学信仰、**或**隶属于哪个工会；**
- 可以**唯一识别一个人的生物遗传学数据。**

个人数据匿名化

- **个人数据匿名化，即无法通过数据集识别数据主体。**这是一个不可逆的过程。数据匿名化后，这些数据不再被视为个人数据，并且GDPR条例不再适用。
- 通常情况下，**建议您不要将原始数据集匿名化。**因为，匿名数据集一定是经过匿名化处理的。数据匿名化后，您无法通过以下方式识别数据主体：
  - 个人数据提取：无法提取与某个数据主体相关的部分或全部记录；
  - 个人数据关联：无法将数据集中与同一个数据主体或群体相关的记录关联起来；
  - 个人数据推理：无法根据数据集内部或外部信息推断出一个人的某项属性。
- 也就是说，在大部分情况下，**这样的匿名化处理会导致数据集质量下降。**G29匿名化技术意见书描述了当今主流匿名化技术，以及被错误视为匿名数据集的案例。需要注意的是，个人数据匿名化迄今没有通用的解决方案。是否对数据进行匿名化以及匿名化技术的选择需要根据使用背景和需求（数据的性质，数据的作用，个人数据泄露风险，等等）来确定。

个人数据假名化

- **个人数据假名化实际上是原始数据和匿名数据之间的折衷。**
- **个人数据假名化，即，在不借助额外信息的情况下，无法识别数据主体。**GDPR条例规定这些额外信息必须单独保存，并采用相关技术和组织措施，避免通过这些信息重新识别出数据主体。与个人数据匿名化不同的是，数据假名化是一个可逆的过程。
- 实际操作中，假名化处理实际是**用间接身份标识数据（别名，某种分类编号等）替换数据集中的直接身份标识数据**（例如姓名），以降低数据的识别性。这些间接身份标识数据可以通过对个人数据进行哈希加密获得，例如IP地址、用户名、电子邮件地址。
- **假名化后的数据仍然被视为个人数据，因此受GDPR条例的约束。**综上各种数据处理方式，欧洲法规较推荐数据假名化处理。GDPR条例认为个人数据假名化可以降低数据主体信息泄露的风险，也更有利于合规。

第二章：开发前准备

从项目设计阶段开始，就必须将保护个人数据的理念纳入到IT开发中，以便时刻保护数据主体的隐私，为他们提供更好的数据管控，并防止应用程序中出现个人数据错误，丢失，未经授权的修改或个人数据滥用。

选择方法

- 通过采纳Privacy By Design 的方法，**将隐私保护置于开发的核心位置。**
- 如果您使用敏捷开发方式，**请将数据安全置于整个流程的核心位置。**法国国家信息系统安全局（ANSSI）提供了“数字安全性和敏捷性”指南，教您如何在敏捷开发的同时保证数据安全。查阅该指南，您一定可以从中获益。
- 对于面向大众的开发项目，需要重点关注与隐私相关的设置，尤其是默认设置，例如，默认可见的用户特征和内容。
- 实施数据保护影响分析（DPIA）。对于某些数据处理操作，必须执行DPIA分析。其他情况下，也非常建议您采用DPIA分析，以便在开发上游，就识别和规避一切风险。CNIL网站上有专门介绍DPIA分析的章节，并提供执行这类分析的免费软件。

选择技术

功能和架构

- **从应用程序或服务的设计阶段开始，就将隐私保护和数据安全的相关要求纳入其中。**这些要求会对程序架构（例如，分布式或集中式）和功能（例如短时间内的匿名化，数据收集最小化）的选择造成影响。应用程序的默认设置必须满足数据安全的基本要求并且符合法律法规。例如，默认的密码复杂度至少须遵循CNIL关于密码设置的相关建议。
- **保持对系统的管控。**时刻对系统进行管控，既可以确保系统正常运行，也可以维护数据安全。使用较为简单的系统，这样您可以明确了解系统内部工作流程，并识别系统的弱点。如果您必须使用一个复杂的系统，建议您从安全的，设计合理的简单系统开始，一点一点增加系统的复杂度，同时确保加入功能的安全性。
- **数据保护不能只有一条防线。**尽管您已经采取了一切措施来设计一个安全的系统，加入的某些组件也可能存在安全隐患。为了将用户数据损害的风险降到最低，还需要对系统进行深度防御。例如，控制在线输入的数据是外围设备防御的一种方式。如果输入数据不合法，请求就会被拒绝。

工具和实践

- **采用考虑安全性的编程标准。**通常，已经有一些改善开发安全性的标准，案例以及编程指南。除此之外，一些辅助工具也可以集成到您的集成开发环境（IDE, Integrated Development Environment ）中。这些辅助工具可以自动检测您的代码是否符合标准和指南中的规定。您可以在网上找到相关编程语言的操作规范，例如，点击这里，查看C，C ++或Java的操作指南。如果是开发网络应用程序，您可以去OWASP网站上查找相关操作，方法和工具。
- **技术的选择至关重要，**须注意以下事项：
  - 根据应用程序的所属领域和功能，选择更适用的编程语言或技术。
  - 经过验证的编程语言和技术更加安全。因为这些编程语言和技术中的漏洞通常已经被发现和修复。但是，当使用各技术模块的最新版本时，您仍需要提高警惕。
  - 避免使用刚刚入门还尚未精通的编程语言来编写解决方案。因为缺乏经验将导致更高的安全风险。
- 请参考该指南中的相关章节，**搭建一个安全可靠的、代码易读的开发环境。**

第三章：保护开发环境安全

搭建安全的开发环境，需要优先保护生产、开发和持续集成服务器以及开发人员工作站的安全，因为这些服务器和开发人员需要大量访问数据。

评估风险并采取措施

- **评估开发流程和工具的风险。**列出您现有的安全措施，然后制定一个计划来提高风险的覆盖率。并指定一个专门人员负责计划的制定和执行。
- 您使用的所有工具都要考虑风险，尤其是 SaaS（Software as a Service）工具和协作工具（例如，Slack、Trello、GitHub等）。

保护服务器和工作站安全

- 关于如何保护**服务器，工作站和内网安全**的建议请参见CNIL发布的《**个人数据安全指南**》中的第5-8章节。
- **撰写文档对这些安全措施进行归类，并阐述各项措施的配置，**以确保各项安全措施在服务器和工作站上统一部署。您可以使用**配置管理工具**，例如 Ansible、Puppet或Chef来简化工作量。
- 条件允许的情况下，建议您让服务器和工作站自动更新。您可以制定一张检查表，罗列所有的高危漏洞，以便对您进行提醒。您可以参照CERT-FR（法国计算机应急响应组织）网站上的安全告警，安全建议，以及安全公告进行制定。

重点实施访问管理和操作追踪

- 记录下SSH密钥（采用最新的加密算法和密钥长度，密码保护密钥，密钥旋转）的管理过程，您可以参考有关安全使用（开放）SSH的文档，获取操作示例。
- 建议对开发团队涉及的业务进行**强身份验证。**
- **追踪**您设备的访问情况，如有可能，实施**日志的自动分析。**为了保证追踪记录准确，应避免使用通用账号。

第四章：管理源代码

无论项目规模大小，都建议您使用源代码管理工具，对源代码的版本进行实时追踪。

安全高效地设置源代码管理器

- 源代码管理器用于存储**所有的源代码和与之相关的程序文档**，同时可保存所有的**修改记录。**简单的FTP服务器不是源代码管理器。

- 使用源代码管理器提供的功能对您的开发环境进行正确设置。强烈建议您在项目开始时，就使用**强身份验证**或SSH密钥认证。
- 此外，您还可以给源代码管理器的用户设置项目访问级别，并为每个级别定义相应的**权限**（例如"访客"具备只读权限，"开发人员"具备读写权限）。
- 对您的源代码管理系统进行**定期备份**，尤其是保存修改记录的主要服务器。您可以采用一个高效的开发流程，即使**多个开发人员同时工作**，也能保证项目顺利进行。例如，多个开发人员不在同一个分支（master分支）上操作，而是按照功能，在不同的分支上进行，然后，随着开发过程的进行，将对应的分支一点一点地合并到主要分支上。类似这样的开发策略在Git Flow中已有详细说明。此外，一些源代码管理器还支持配置"**受保护的分支**"，用来防止对这些分支中的文件进行非法篡改。

注意源代码内容

- 使用**代码质量测量工具**。一旦代码被提交，该工具会立即对代码进行扫描，检验代码质量。您也可以在配置源代码管理器时添加代码质量测量的控制脚本：源代码质量不达标，代码无法提交。
- 在源代码仓库外保存您的密钥和密码：
  - **单独保存在无需提交的文件中。**保存时，使用源代码管理器的特殊文件格式（例如，Git 的 gitignore 文件格式），避免误提交这些文件。
  - **保存在环境变量中**，并确保当应用程序出错时，环境变量不会被意外显示或写入日志中。
  - 使用专用的密钥管理软件或配置管理软件。

如果您必须将密钥和密码保存在源代码仓库内，请使用源代码管理器插件（例如git-crypt）**自动对文件进行加密和解密。**

  - 如果您提交了包含个人数据或其他关键信息的内容，请务必将其从源代码仓库中彻底删除，因为即使修改后，仓库历史记录中的相关数据仍然可用。
- 源代码上线需谨慎。**上线前**，需仔细检查代码的**全部内容**，确保代码以及修改记录中没有任何个人数据，密码或其他密钥。

工具示例

- 与Subversion这类需要中心服务器的工具不同，主流源代码管理器（例如，Git 和 Mercurial）都是**分布式**的。
- 这些工具中大部分都提供**网页界面和一些辅助工具**(例如，bug管理工具、wiki文档编辑工具等)。上述工具既可以在线使用（GitHub, Bitbucket），也可以安装在内部服务器上（GitLab）。

第五章：合理选择架构

在设计应用程序架构时，需要识别收集的个人数据，对这些数据的去向进行分类，并为不同去向的数据设置保留期限。数据存储介质的选择非常关键，不仅要选择您所熟知的存储介质，还要与您的需求相匹配。您可以根据数据处理记录表和数据保护影响分析来选择数据存储介质。

分析数据去向

- 在项目开发上游，就规划好项目的预期功能，并绘制数据流图，详细描述要执行的流程和使用的数据存储介质。
- 当数据只存储在**用户终端设备**上（本地存储），或限制在**用户可控的通信网络**（例如，Wi-Fi或其他本地网络）上时，须重点注意数据安全。用户可以自行决定这些数据的保留期限，也可以随时删除这些数据。
- 当**数据通过网络业务进行传输时**，自行托管这些数据还是交由供应商托管，需要根据您对安全情况的判断以及所期望达到的业务质量来决定。云托管虽然公认具有更高的安全性，但是同样存在其他风险。您可以查阅CNIL关于云计算的建议获取相关指导。

第三方数据托管

- **选择一家流程透明的，并具备隐私和安全保护措施的托管服务提供商。**为此，CNIL提供相关隐私条款示例供您查阅。
- **确保您了解数据托管服务器的地理位置。**您可能需要在欧盟（EU）和欧洲经济区（EEA）以外进行数据传输。如果数据可以在欧盟（EU）或欧洲经济区（EEA）内部自由传输，则在确保数据被充分和恰当保护的前提下，可以在上述空间外进行传输。 CNIL提供相关地图，用于查看不同国家的数据保护程度。
- **如果您需要托管服务提供商来托管健康状况数据**，请务必确保该提供商对托管此类数据具备认证和许可。
- 其他注意事项如下：
  - 托管服务提供商具备可访问的安全策略。
  - 托管服务提供商具备相关措施保证托管站点的硬件安全。
  - 具备数据加密或其他流程确保托管服务提供商无法访问被托管的数据。
  - 更新管理，授权管理，操作人员身份验证以及应用程序开发安全性。
  - 结构化数据存储，随时按需可逆和可迁移。

第六章：保护网站、应用程序和服务器安全

任何网站，应用程序或服务器都需要将最新的安全准则集成到通信、身份验证以及基础设施中。

保护通信安全

- **所有网站均采用1.2或1.3版本的TLS协议**（而不是SSL协议）来传输移动应用程序数据，例如通过LetsEncrypt获取TLS证书。请使用最新版本的TLS协议并确保该协议正常运行。
- **强制网站上的所有页面和移动应用程序使用TLS。**
- 在保证应用程序正常运行的情况下，**使用最少的通信端口。**如果只能通过HTTPS协议接入网络服务器，则只有端口443和80可以开放，其他端口都必须已被防火墙关闭。

- 法国国家信息系统安全局（ANSSI）网站上发布了关于部署TLS协议和保护网站安全的相关建议，供您查阅。

保护身份验证安全

- **查看CNIL关于密码设置的建议。**用户访问时，限制其输错密码的次数。
- **切勿以明文形式存储密码。**应将密码通过哈希加密形式存储在可靠的数据库中，例如bcrypt加密功能。
- **使用cookies进行身份验证时**，建议您
  - 开启HSTS让浏览器强制跳转HTTPS访问
  - 使用安全标志；
  - 使用HttpOnly标志。
- **测试系统上安装的密码套件**，并停用旧的密码套件（RC4、MD4、MD5等）。我们建议您使用AES256加密算法，详细内容请参考ANSSI相关说明。
- **管理员需使用特定的密码策略。**当更换管理员，或怀疑存在数据泄漏时，需要更改密码。尽可能使用高强度的密码。
- **仅允许授权人员访问管理工具和界面。**如果是日常操作，建议使用普通权限的账户登录。
- **如果从Internet访问管理界面，需采用高强度的安全措施。**例如，在内部服务器上部署对用户及其使用的站点进行强身份验证的VPN。

保护系统基础设施安全

- **对系统进行备份，如有可能，将备份数据加密并定期检查。**尤其是当您的系统遭到勒索软件攻击时，您只能通过整个系统的备份来恢复系统。
- **限制系统中运行的组件数量，**并且每个组件须执行以下操作：
  - **每周自动检查，**以便及时安装重要更新；
  - **自动进行漏洞监视，**例如，通过订阅CERT-FR安全公告。
- **使用漏洞检测工具**对核心的数据处理操作进行检查，以便及时发现潜在的安全漏洞。您也可以在关键的系统和服务器上安装攻击检测预防系统。上述检查需定期进行，并确保在新的软件版本投入使用之前进行。
- **限制或禁止对诊断和配置端口进行远程的物理和逻辑访问。**例如，您可以使用netstat工具列出所有开放的端口。
- **保护可以通过Internet访问的数据库**，您可以限制访问权限（例如，IP筛选）并更改管理员帐户的默认密码。
- 在数据库管理方面，注意事项包括：+
  - **使用名义帐户（非通用账户）**访问数据库并为每个应用程序创建专有的帐户；
  - **撤销名义帐户或应用程序帐户的管理权限**，以防应用程序数据库结构被篡改（表、视图、过程等）；
  - 通过嵌入SQL代码或脚本来实施防攻击措施。
  - 对硬盘和数据库中的数据进行加密。

第七章：数据收集最小化

您仅应收集恰当的、相关的，且仅限于处理目的所必需的个人数据。

收集之前，考虑要收集的数据类型，并限制在严格必要范围内

- 在应用程序开发之前，考虑好要收集的**数据类型**，并**记录下来**。
- 如果某种数据**对于某类用户来说不是必需的**，就不要收集这些数据。
- 对数据进行处理，**降低数据的精确性**（例如，个人数据假名化），然后存储数据。例如，如果应用程序只需要用户的出生年份，则仅存储出生年份而不是完整的出生日期。
- 在收集敏感数据时，例如与**健康状况或刑事处罚**相关的数据，请确保**数据收集最小化**。出于法规限制，这些数据**能不收集就不收集**。
- 数据收集最小化同样适用于**日志数据**，并且不要将敏感数据或关键数据（健康状况数据、密码等）存储在日志中。
- 某些功能可以改善用户体验，但是对于**您应用程序的正常运行不是必需的**（例如，定位功能使得用户寻找某一地点更加方便）。这种情况下，**必须允许用户选择是否使用此功能**。如果用户选择使用，则该功能收集的相关数据在执行该功能所需的时间内保留，并且严禁用于其他目的。
- 根据数据处理的目的以及数据保留的法律法规，为每个数据类型设置**保留期限**。日志数据同样需要设置保留期限。记录下您设置的保留期限。对于各保留期限，您都必须有足够的理由证明其合法性和正当性。

数据收集后，设置自动删除机制

- 当数据保留期限到期时，设置**自动删除**系统将到期数据删除。您还可以定期对存储的数据进行手动检查。
- 为确保完全清除到期数据，请您**物理删除**所有不需要的数据，您可以使用专用工具或直接将物理存储介质清空。
- 如果数据对您来说仍然有用，您可以通过数据**假名化甚至匿名化**来降低数据敏感性。进行假名化处理后，这些数据仍受个人数据保护条例的约束（请参见第一章节）。
- 用日志记录**自动删除过程**。这些日志可作为您**删除数据的证明**。

第八章：管理用户

在开发上游，就计划好如何管理您的合作人员和最终用户的画像。即根据不同角色定义不同的访问和授权条件，以便每个人只能访问他们实际需要的数据。

用户管理操作指南

- 首先，无论是应用程序用户还是开发合作人员，**每个人都必须使用唯一的账号。**
- 按照CNIL的建议，在访问个人数据之前必**须进行身份验证。**
- 为确保每个人（应用程序用户或开发合作人员）只能访问他们**实际需要的数据**，您的系统需要根据需求和使用者设计**差异化的数据访问管理策略**（读取、写入、删除等）。采用全局的用户画像管理机制，根据每组用户在应用程序中发挥的作用，为每组用户分配权限。
- 用户画像管理可以配合**日志系统使用**(即，将相关记录写在"日志文件"或"日志"中)，以便于**追踪用户活动并检测与之相关的异常或事件**，例如欺诈性访问和滥用个人数据。这些机制除确保用户正确使用信息系统外，严禁用于其他目的，并且日志保留不能超过规定期限。上述日志系统不能超出数据保留期限保存数据。通常的合理期限为6个月。
- 您还可以在开发环境中实施代码审核或渗透测试，以确保**用户画像管理系统的可靠性。**

流程化授权画像管理

- **在文档中记录**开发合作人员的活动轨迹和用户的注册退订操作或使**该管理流程自动化。**例如，上述流程应包括发现无权限人员访问某站点或IT资源，或合同到期时，采取的相应措施。
- 在用户和合作人员管理过程中，随着项目内部人员变动和项目使用情况的变化，**需定期检查授予的权限。**建议您使用轻型目录访问协议（LDAP、*Lightweight Directory Access Protocol*）之类的目录服务，以便您及时跟进这些变化，并细化访问策略。例如，根据使用画像为不同用户分配权限，这样您既可以满足用户访问需求，又能将赋予的权限最小化。
- **对于常规操作，应避免使用"高级"帐户**（root类型账户、管理员账户等），因为高级账户是系统的基石，也是外部攻击的首选目标。建议您对这类账户使用强密码策略（10到20个字符的密码或多因素身份验证），并仅允许必要的人员知晓该密码。
- **建议在项目中使用密码管理器。**如有可能，尽量使用高强度的密码。同时，也应避免多人使用一个通用帐户。

第九章：管理您的库和SDK

您是否使用第三方的库，软件开发工具包（SDK）或其他软件组件？以下是一些建议，教您如何在安全开发的同时集成这些工具。

做明智的选择

- **考虑是否有必要添加依赖项。**虽然一些常用软件模块的实现只需要几行代码，但是添加的每个元素都会扩大系统的攻击面。在使用单个库实现多种功能的情况下，仅集成您实际需要的功能，也就是说，使用的功能越少，越能减少潜在的bug数量。
- **选择在维护中的软件，库和SDK：**

- 如果您要使用免费或开源软件，尽量选择具备活跃社区和完善文档并定期更新的项目或解决方案；
- 如果您使用其他需要付费的解决方案，请通过合同确保在项目的整个生命周期内，该第三方都对其代码进行维护和更新。

- **考虑隐私问题。**某些SDK或库利用从应用程序或网站中收集的个人数据来获取利益。请您确保此类第三方遵守有关个人数据的现行法律，并确保您的应用程序或网站提供相应机制来征求用户同意。
- **如果您需要使用加密机制，强烈建议您不要自己实施加密算法或协议，**而要选择时常维护的，公认的和易于使用的密码库。

评估所选依赖项

- **阅读您所选依赖项的文档资料，并修改其默认配置。**了解依赖项的工作原理。第三方库和SDK通常带有默认配置文件，但是这些文件由于时间不足而很少修改，易导致安全漏洞。
- **检查您的库和SDK。**您真的知道您使用的所有库和SDK的功能吗？这些依赖项发送哪些数据？这些数据发给谁？通过检查，您就可以明确在数据保护方面应遵守的义务，并明确相关参与者的责任；
- **映射依赖项。**第三方库和SDK也可以集成其他组件：检查第三方库和SDK的代码帮助您更好地映射所有依赖项并在某个依赖项出现问题时采取行动。也建议您对集成的第三方组件进行安全检查和监视；
- **提防误植域名和其他恶意技术。**检查依赖项的名称以及依赖项集成的组件，以防攻击。不要从未知站点复制粘贴命令行。

维护库和SDK

- **使用依赖项管理系统**（例如yum、apt、maven、pip等）便于您时刻了解这些依赖项。
- **管理依赖项更新，**尤其是修复漏洞的安全更新。并将依赖项更新的管理和部署过程记录在程序文档中。
- **过时的库和SDK版本将不再维护**：因此您需要寻找其他解决方法（选择新的库、购买更新的商业版本）；
- **监视开源项目的状态，**谨防域或程序包的所有者发生变化，或依赖项恶意更新攻击。

第十章：注意代码和文档质量

保持代码清晰整洁非常重要。良好的代码可读性将减少您，您的合作人员或今后的修改人员的工作量，包括维护，审核和修改错误。

代码和架构文档化

- 在开发过程中，由于时间有限或缺乏对项目的整体了解，撰写对程序功能进行解释说明的文档通常被您置之脑后。然而，**程序文档对于提高项目的可维护性至关重要**：撰写这类文档有助于您全面了解代码的功能，并知道代码的哪些部分会受到修改的影响。
- **撰写文档解释项目架构，而不仅仅是解释代码：**撰写文档时，需保持全局视角，从而帮助其他开发人员对各

组件的运作有一个整体认知。因此，在撰写项目文档时，建议您使用清晰的图表和说明。

- **随代码的变化同步更新文档。**最好的方法就是，在修改代码时，同步修改文档。
- **随代码的变化，将文档版本化。**如果您使用源代码管理器，您可以在每次提交代码修改时，也一并提交对文档的修改。（请参见章节《管理源代码》）。
- **注意保护文档的安全性**：考虑用户或开发人员文档安全方面的问题。为此，您可以用文档记录下您应用程序可以选择的不同配置，并说明哪些设置最安全。

控制代码及文档的质量

- 在整个程序编写过程中，**既要采用合规的操作方法，也要严格遵守编码约定**，才能获得高质量的代码。您最好参考现有的约定来选择您的编码约定。以下是操作示例：
  - **使用含义明确的变量名和函数名，**让别人乍一看，就了解代码的大致功能。
  - **正确缩进代码让代码层级结构更加清晰。**
  - **避免代码冗余**，可以让您在修改某功能时，减少需要修改的地方。因为修改的地方越多，越容易遗漏。
- **一些工具可以控制代码质量。**正确设置这些工具，可避免重读代码来检查是否符合编码约定。
- **以下工具可以帮助您控制代码质量：**
  - 使用插件，对**集成开发环境（IDE）**进行设置，使之遵守代码缩进、换行以及大括号或其他括号的使用规则。
  - **源代码质量测量软件**可以报告代码重复率，是否符合编程规则或潜在的bug。

第十一章：测试您的应用程序

对您的产品进行测试，以确保其运作正常、用户体验良好，并且在投入生产前后均不出现故障。测试产品还有助于降低个人数据受损的风险。

测试自动化

- **开发测试**（单元、功能等）将检查产品规格和功能之间的一致性。**安全测试**（随机数据测试，又名"模糊测试"、漏洞扫描等）将测试该产品在我们对其进行非正常使用时是否能够继续正常工作，并且不存在可以让第三方损害其安全性的漏洞。这两种测试对保证您应用程序正常运行至关重要。
- 搭建一个**持续集成系统**，以便每次修改源代码后自动启动测试。

将测试纳入到您的企业策略中

- 在企业策略中搭建测试环境时，必须在开发之前，由各方共同定义**指标阀值。**
- 要考虑的指标，示例如下：

- 测试的**覆盖率**及测试类型；
- 代码**重复率**；
- **漏洞数量**（工具所检测到的）及漏洞类型等。

谨慎使用测试数据！

- 在开发和测试阶段**不要使用**"真实的"生产数据，而是使用来源于生产库的测试数据，以便**与真实数据区分开。**
- 在生产用途之外使用个人数据时，**安全风险会增加**，例如，没必要查看数据的人访问了数据，存储地点的增加等。
- 因此，您需要构建一个**虚拟数据集**，该数据集与您应用程序将要处理的数据类似。虚拟数据集中的数据泄露不会对数据主体造成影响；
- 如果您需要从生产环境中**导入现有的配置**到测试环境中，请将可能存在的**个人数据匿名化。**

第十二章：告知用户

GDPR条例的透明性原则要求与个人数据处理相关的任何信息或告知必须简明、透明、可理解且易于获取，并且告知措辞简单而清晰。

告知谁？何时告知？

- 须在以下情况告知用户：
  - **直接收集个人数据**，即直接向受众收集数据（例如：填写表格、网上购物、签订合同、开通银行帐户）或通过观察受众活动的装置或技术收集数据（例如：互联网浏览分析、地理定位和Wi-Fi分析/追踪用于访问量统计等）；
  - **间接收集个人数据**，即数据不是向受众直接收集的（例如：从业务合作伙伴、数据中介、可公开访问的资源，或其他人那里收集到的数据）。
- 须在以下时刻告知用户
  - **直接收集个人数据时**；
  - **间接收集个人数据时，应尽快**（例如，与该用户首次联系时）告知用户，或最迟在一个月内（特殊情况除外）告知用户。
  - **实质性修改或特定事件发生的情况下。**例如：更改数据使用目的，更改数据接收人，权利行使方式的变化或个人数据受损。

告知用户什么?

- 任何情况下，您都必须说明：
  - 收集数据的机构的**身份和联系方式**（谁处理数据？）；
  - **目的**（收集的数据将用于什么目的？）；
  - 数据处理所基于的**法律依据**（在法律依据中查找相关信息）；
  - **数据收集的强制性或可选性**（假定数据收集的必要性，按照“数据收集最小化”原则进行判断）以及在不提供该数据的情况下，**会给数据主体造成什么样的后果**；
  - **数据的接收者或接收者类型**（鉴于定义的目的，谁（包括供应商）需要访问或接收数据？）；
  - **数据保留期限**（或确定保留期限的原则）；
  - **数据主体拥有权利，并且有方法行使这些权利**（数据主体的访问权、纠正权、擦除权和限制处理权适用于任何数据处理）；
  - 机构的**数据保护官**（如果已任命）或数据保护相关单位的**联系方式**；
  - **向CNIL委员会投诉的权利。**
- 在某些特定情况下，还必须提供额外信息，例如，在欧盟以外进行数据传输，自动决策或用户画像决策，收集数据的机构出于追求其合法利益而处理数据（更多信息，请参见CNIL网站）。
- 如果是间接收集，还须告知用户以下信息：
  - 收集的**数据类型**；
  - **数据的来源**（须特别指出数据是否来自可公开访问的资源）。

以何种形式告知?

- 告知信息应该**易于获取**：用户可以毫不费力地找到告知信息。
- **告知信息要清晰易懂**，即使用简单的词汇（句子简短，没有法律或技术术语，没有歧义）和适合目标受众的信息（尤其需要注意儿童和弱势群体）。
- **告知信息要简洁。**为了避免信息泛滥给用户带来困扰，**必须在合适的时机提供最相关的信息。**
- 必须将隐私保护的相关信息**与其他信息区分开**（例如，在合同条款或一般使用条款中）。

数据安全受到损害时，如何告知用户?

- **数据收集机构有时会由于失误或疏忽，导致数据遭受意外或恶意的破坏，即数据安全受到损害。例如，数据破坏，数据丢失，数据篡改或未经许可的披露。**在这种情况下，如果数据安全损害可能会给数据主体的权利和自由带来风险，相关机构必须在**72小时内**向CNIL报告违规行为。
- 如果风险很高，相关机构还必须尽快将情况告知数据主体，并为他们提供数据保护的建议（例如，注销受损的银行卡、修改密码、修改隐私设置等）。
- 请通过CNIL网站向CNIL报告违规行为。

相关资源

- 由CNIL数字创新实验室(法语：Laboratoire d'Innovation Numérique)开发的站点"数据与设计"（法语：Données & Design）阐述了上述概念，并为您提供告知用户的界面示例。
- CNIL网站上还为您提供大量信息告知案例。
- CNIL网站上"损害个人数据"页面。

第十三章：数据主体行使权利

您处理其数据的数据主体拥有对该数据的权利：访问权、纠正权、反对权、擦除权、数据可携带权和限制处理权。

您必须为数据主体提供有效行使其权利的方法，并在您的计算机系统中提供允许其行使权利的技术工具。

在开发上游，就计划好数据主体如何与您联系以及您如何处理他们的请求，以便您对上述权利的行使进行有效管理。

基本措施

- 任何使用个人数据的组织机构都有义务告知数据主体**在何处以及如何**行使与该数据相关的权利。例如，在告知数据主体时，以及在隐私政策中提供您的邮箱地址或征求用户同意的表格。
- 为了方便数据主体行使权利，这些权利也可以全部或部分**在您的应用程序或软件中实现。**这种实现不是强制的，但可以满足用户需求，并减少处理这类请求的时间和复杂性。
- 最重要的是，在数据主体为行使其权利而直接访问和操作数据的情况下，切记以安全的方式管理其**身份验证。**一般来说，还要追踪对其个人数据造成影响的一切操作。

权利在程序中的实现示例

- **访问权**：数据主体有权获取您拥有的与其相关的所有信息的副本，这可以让数据主体知道是否处理了与他相关的数据，并获得格式内容均可理解的副本。同时，也让数据主体可以控制其数据的准确性。

  可能的实现方法：提供一种功能来显示与数据主体相关的所有数据。如果数据很多，可以将这些数据分成多个部分进行显示。如果数据量庞大，须为数据主体提供可下载的包含其所有数据的压缩包。

- **擦除权**：数据主体有权要求删除您拥有的与其相关的所有数据。

  可能的实现方法：

  1. 提供一种功能来清除与数据主体相关的所有数据。
  2. 须提供供应商自动通知机制，以便供应商删除与此数据主体相关的所有数据。
  3. 提供在备份数据中删除数据的机制，或其他解决方案，以确保删除的数据无法恢复。

- **反对权**：在某些情况下，数据主体有权反对将其数据用于特定目的。

可能的实现方法：提供一种功能，允许数据主体反对进行的数据处理。当数据主体以此方式行使其反对权时，数据处理负责人必须删除已收集的数据，并且今后不得再收集与该数据主体相关的数据。

- **数据可携带权**：数据主体有权以机器可读的格式取回其数据，供自己使用或提供给其他机构。

  可能的实现方法：提供一种功能，允许数据主体以计算机可读的标准格式（CSV、XML、JSON等）下载其数据。

- **纠正权**：数据主体有权要求对不正确的数据进行修改，以限制错误信息的使用或传播。

  可能的实现方法：允许用户直接修改其帐户中的数据。

- **限制处理权**：数据主体有权要求在一定时间内停止对其数据进行处理。例如，用户对数据使用提出异议，审查该争议的时间内，或数据主体要求行使权利的时间内。

  可能的实现方法：允许管理员将该数据主体的相关数据暂时"隔离"：无法再读取或修改这些数据。

总结

- 由CNIL数字创新实验室（法语：Laboratoire d'Innovation Numérique）开发的网站"数据与设计"（法语：Données & Design）阐述了上述概念，并提供允许用户行使权利的界面示例。
- 最后，在实现方法上发挥您的**创造力**吧！（如有疑问，请向CNIL寻求建议。）

第十四章：数据保留期限

个人数据不能无限期保留：必须根据处理目的进行定义。一旦目的达到，就应将这些数据存档，删除或匿名化（例如，用于进行统计）。

数据保留周期

- 个人数据的保留周期**依次分为三个阶段**：
  - 活跃数据库；
  - 中间阶段存档；
  - 最终存档或删除。
- 从活跃数据库中删除个人数据的机制可以确保相关操作业务仅在**实现处理目的所必需的时间内**保留和访问数据。
- **注意不要将数据直接保留在活动数据库中**，而是将其**标记为已存档**。存档的数据（中间阶段存档）仅应被负责访问该档案并将其从该档案中删除的特定业务访问。
- 注意为存档数据设置**特定的访问条件**，因为数据存档必须是及时的和谨慎的。
- 如有可能，在**数据删除或匿名化**时，使用与**擦除权**相同的实现方式（请参见《数据主体行使权利》章节），以确保您的系统功能一致。

数据保留期限示例

- **与薪资管理或员工工作时间控制相关的数据**可以保存5年。
- **医疗文件中的数据**必须保存20年。
- **不响应任何请求的潜在客户的联系方式**可以保存3年。
- **日志数据**可以保存6个月。

第十五章：考虑技术实施中的法律依据

个人数据处理必须基于GDPR条例第6条所提及的"法律依据"之一，方可进行。也就是说，数据处理的法律依据是数据处理存在的理由。法律依据的选择将直接影响处理条件和数据主体的权利。因此，在开发上游，就要为项目中涉及的数据处理提供法律依据，并在项目中集成必要的功能，以确保遵守数据处理的法律法规和尊重数据主体的合法权利。

GDPR法律依据的定义

- 在私营组织（公司、协会等）发展背景下，最常用的法律依据是：
  - **合同**：数据处理是数据主体与执行处理的组织之间履行或准备合同所必需的；
  - **合法利益**：实施处理的组织为追求其"合法"利益而处理数据，并且此处理不影响数据主体的权利和自由；
  - **同意**：数据主体对处理表示明确同意。
- 如果您是公共权力机关或维护公共利益的组织，则还可以使用以下法律依据：
  - **法律义务**：处理是法律文件强制的。
  - **维护公共利益的使命**：处理是保护公共利益所必需的。
- CNIL网站为您提供了相关案例，帮助您选择适用的法律依据。
- 最后，在非常特殊的情况下，**保护生命安全**可以作为法律依据。例如，追踪流行病的传播或采取人道主义应急措施的情况下。
- 您需要先在CNIL网站上确认您的情况不涉及法律规定的**特殊限制**。（例如：向用户发送广告邮件、某些Cookie和其他追踪程序等）。

选择适当的法律依据

- **对于一个给定的处理目的，只能选择一个法律依据。**对于同一目的，法律依据不能累积。同一种数据处理可以达到多种目的，即有多个目标，必须分别为每种目的定义法律依据。
- 如上所述，如果您是**公共组织**，则大部分情况下，履行法律义务和维护公共利益是最适当的法律依据。
- 如果您的处理是合同关系的一部分，且是为用户提供服务客观上所必需的（例如，用户在购物网站上创建账户时填写的姓名和地址），**则履行合同是最适当的法律依据。**

- 如果您的处理不属于与用户合同关系的一部分，**则用户同意或追求合法利益可以作为最适当的法律依据。**如果您的处理可能具有侵扰性（用户画像，地理定位数据的收集等），则用户同意可以作为最适当的法律依据。
- 当您的处理包含**敏感数据**（健康状况数据、与性生活或性取向相关的数据等）时，除了法律依据外，您还必须识别GDPR第9条规定的例外情况。

根据选择的法律依据告知用户并提供行使权利的方法

- 根据您的情况，填写法律依据和对应的用户行使权利的方法。

	访问权	纠正权	擦除权	限制处理权	数据可携带权	反对权
用户同意						**用户撤销同意**
合同						
合法利益						
法律义务						
公共利益						
生命安全						

- 所采纳的法律依据必须**始终出现在告知数据主体的信息中。**
- **当您的处理基于合法利益时**，您还必须指出所追求的合法利益（打击欺诈、维护系统安全性等）
- **建议记录您选择的法律依据。**例如，选择的法律依据可以记录在数据处理流程图中或程序文档中。

Cookie和其他追踪程序

- 欧洲ePrivacy法令要求，在通过cookie，账户或其他追踪程序（软件指纹、像素）存储用户信息或访问存储在用户终端设备中的信息之前，须征求用户同意。
- 然而，也存在例外。允许或方便电子信息交流的cookie或为用户提供其要求的服务所严格必需的Cookie，可免除同意。
- 在某些情况下，与访问量统计相关的cookie也可免除同意。
- 另外，将单个追踪程序用于多个目的并不能免除针对其他目的的同意。例如，如果身份验证cookie也用于广告推送，则必须针对后一个目的征求用户同意，征求同意的方式与未"登录"的网站相同。

第十六章：网站和应用程序访问量统计

访问量统计工具用于获取相关网站或应用程序用户的浏览信息。这些信息有助于了解用户登录方式以及重现用户浏览路径。除特殊情况外，对cookie的使用都应遵从用户同意的原则。

获得用户同意

- **使用cookie或者追踪程序之前**，网站或应用程序的开发人员必须遵守以下规定：
  - 告知用户获取cookie的意图；
  - 获得用户同意；
  - 告知用户拒绝使用相关cookie的方式。
- 除非符合下文定义的情况，**否则征求同意的义务适用于所有统计访问量的追踪程序。**

免除同意条款

- **在满足一定条件的情况下**，用于访问量统计的cookie可免除同意。
- **这些条件**在**关于cookie和其他追踪程序准则**中有所规定，**如下**：
  - 告知用户使用情况；
  - 赋予用户反对的权利；
  - 将上述装置权限限定在以下范围：
    - 用户访问量统计
    - A/B测试
  - 不交叉获取其他处理操作的数据（例如客户文件、其他站点的流量统计等；）
  - 将追踪程序的范围限制在单个网站或应用程序中；
  - 删除IP地址的最后一个八位位组；
  - 追踪时间不超过13个月。
- 只要满足相关条件，就能**从Opt-in（选择进入）状态转向Opt-out（选择退出）状态。**
- 此外，同一个第三方（供应商）可以向多个网站或应用程序开发商提供用户访问量统计服务，条件是**每一个开发商独立地收集，处理和存储用户数据，并且追踪程序也彼此独立。**

应用实践

- **当第三方cookie或追踪程序供应商指出，要将您网站或应用程序中的数据重新用于自己的业务时，访问量统计服务不在免除同意范围之内。**对于市场上一些大型的用户追踪服务供应商，尤其如此（请参阅 Google Analytics、Quantcast Analytics 或 Facebook Analytics 的隐私条款）。在某些情况下，可以通过设置上述工具来关闭数据重用，具体信息请向相关服务供应商咨询。
- 如果您想获得免除同意权，请联系追踪服务供应商，或使用可以自行配置的免费软件，例如Matomo（原名Piwik）。