

```
app.MapGet("/trust/complete", async (TrustedDb db) =>
    await db.TrustCode.Where(t => t.IsComplete).ToListAsync());

app.MapGet("/trust/incomplete", async (TrustedDb db) =>
    await db.TrustCode.Where(t => !t.IsComplete).ToListAsync());

app.MapGet("trust/{id}", async (int id, TrustedDb db) =>
    await db.TrustCode.FindAsync(id) is Trusted trust
        ? Results.Ok(trust)
        : Results.NotFound());

app.MapPost("/trust", async (Trusted trust, TrustedDb db) =>
{
    if (!MinimalValidation.TryValidate(trust, out var violations))
        return Results.ValidationProblem(violations);

    db.TrustCode.Add(trust);
    await db.SaveChangesAsync();

    return Results.Created($"/trust/{trust.Id}", trust);
});
```

# TRUST CODE

Microsoft 商业行为准则

```
app.MapDelete("/trust/delete-all", async (TrustedDb db) =>
    Results.Ok(await db.Database.ExecuteSqlRawAsync("DELETE FROM TrustCode")));

app.Run();

Task EnsureDB(string connectionString)
{
    var options = new
        DbContextOptionsBuilder<TrustedDb>().UseSqlite(connectionString).Options;
    using var db = new TrustedDb(options);
    return db.Database.MigrateAsync();
}

enum Courses
{
    Accessibility,
    CompetingFairly,
    DigitalSafety,
    EthicalSelling,
    GlobalReadiness,
    InsiderTrading,
    ModelCoachCare,
    ResponsibleAI,
    Privacy,
    SecurityZeroTrust
}
```



团队成员们：

我们每个人的言行塑造了我们的文化。我们努力打造的是一种多元且包容的文化，我们通过这种文化倡导学习、促进信任的建立，并让每位员工都能出色地完成工作。

在工作中做出正确的决策和合乎道德的选择，有助于我们彼此、我们与客户以及我们与合作伙伴之间建立信任。永远不要为了换取任何短期收益而损害个人诚信或 Microsoft 的声誉和信任。

在制定决策的过程中遵循诚信、诚实和合规的指引，我们就更有可能做出合乎道德的选择。面对困难的处境，我们应始终公开动机，从错误中汲取教训，并寻求帮助。我希望领导层和管理人员能够培养一种能让员工在遇到异常状况时自由提出问题和表达疑虑的文化。

我们的商业行为准则（即，我们的“信任准则”）强调了我们每个人在建立信任方面发挥的作用，以及你在决策过程中应该采取的方法。当我们在日常工作中应用这些原则时，我们将能够满怀信心地在推进工作的过程中制定正确的决策，从而建立信任并帮助客户和合作伙伴成就不凡。

谢谢。

**Satya Nadella**

董事长兼首席执行官



# 目录

## 我们的文化 和价值观

我们的使命

报告问题

成就不凡

管理人员的职责

运用信任准则

直言不讳

## 客户对我们的 信任

尊重隐私

构建安全的产品和服务

请勿进行不当付款

公平竞争

## 政府与社区对我们的 信任

尊重世界各地的法律

尊重并倡导人权

设计无障碍  
产品和服务

开发和部署负责任 AI

## 我们对彼此的 信任

促进多元与包容

为打造安全高效的工作场所做出贡献

避免利益冲突

## 投资者和公众对我们的 信任

勿将内部信息作为交易筹码

保护 Microsoft 的资源

保存准确的记录和合同

保护机密信息和  
知识产权

向公众传达准确的信息

## 我们的代表对我们的 信任

负责任地处理礼品、招待和  
旅行

启用值得信赖的代表

诚信地选择供应商

## 坚持我们的 信任 准则

坚持我们的信任准则



# TRUST

信任我们的文化和价值观

我们的使命

5

成就不凡

6

如何运用信任准则

7

如何报告问题

9

在事情不对劲时直言不讳

10

管理人员的职责

12

```
app.MapGet( "/trust/{id}" , async (
    await db.TrustCode.FindAsync(id)
    ? Results.Ok(trust)
    : Results.NotFound());

app.MapPost( "/trust" , async.(Trusted t
{
    if (!MinimalValidation.TryValida
        return Results.Validation
    db.TrustCode.Add(trust);
    await ChangesAsync();
    return Results.Created($" /trust/"
});
app.MapGet( "/trust/{id}" , async (int
```







# 我们的使命

予力全球每一人、每一组织  
成就不凡。

## 达成我们的使命

### 我们的价值观

尊重

诚信

责任心

### 我们的文化

成长型思维

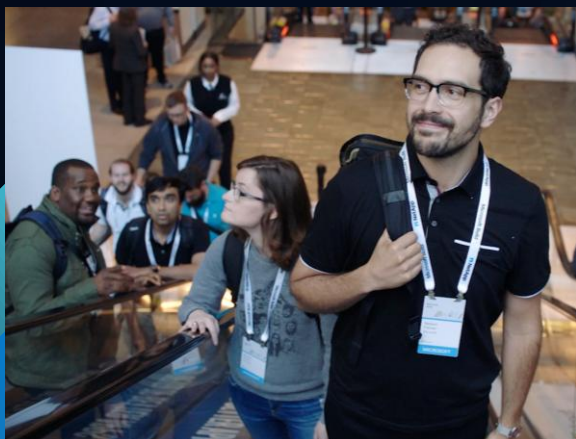
客户导向

One Microsoft

多元且包容

发挥影响力

# 成就 不凡



## 我们如何赢得并 培养信任？

首先运用我们的文化和价值观在  
Microsoft 内外建立持久的关系。

**我们的价值观**是指导我们每天努力  
赢得信任，诚信开展业务的持久原则。

**我们的文化**是我们的运营框架，  
它定义了我们的角色以及我们的  
行为方式。



## Microsoft 商业行为准则

Microsoft 的商业行为准则（即，  
我们的“**信任准则**”）将向你展  
示我们如何运用公司的文化和价  
值观，培养并维系客户、政府、  
投资者、合作伙伴、代表对我们  
以及我们对彼此的信任，从而共  
同成就不凡。

# 如何运用信任准则做出 正确的决策



做出正确的决策和合乎道德的选择有助于我们每个人和那些与我们互动的人建立信任。

但并非你遇到的所有情形都简单明了 - 在面对困难或不明确的情形时，如何做出最佳选择？如何应对道德困境？

虽然我们的信任准则并未告诉你在各种情形下具体该怎么做，但它能够以指南的形式帮助你做出正确的决策，并应对可能并不总是有清晰解决方法的复杂情形。

```
app.MapGet<T>
    .Await
    .ConfigureAwait(
        false)
    .Result;

app.MapPost<T>
    .Await
    .ConfigureAwait(
        false)
    .Result;

app.MapPut<T>
    .Await
    .ConfigureAwait(
        false)
    .Result;

app.MapDelete<T>
    .Await
    .ConfigureAwait(
        false)
    .Result;
```

```
Id, Truste
usted, tr
```





# 当面临困难的抉择或情形时， 按照下列步骤操作：

```
app.MapGet( "/trust/{id}"  
    await db.TrustCode  
    ? Results.Ok  
    : Results.No  
  
app.MapPost( "/trust", a  
{  
    if (!MinimalValidation.TryValidate(trust  
        return Results.ValidationProblem(  
db.TrustCode.Add(trust);
```

Microsoft  
runs on  
trust.

1



## 暂停

这种情形会让你不安吗？你的直觉是否告诉你有些事情不太对劲？

停下来考虑如何处理这种情形。

2



## 思考

你的做法是否符合 Microsoft 信任准则中的文化和价值观？  
你的做法能否培养或维系信任？永远不要为了短期利益牺牲长期的声誉和信任。

3



## 提问

提出问题并寻求帮助。与 CELA、财务或人力资源交谈。  
向你的经理或其他管理人员寻求帮助。  
这些资源可帮助你做出正确的决策 **发送电子邮件至**  
[buscond@microsoft.com](mailto:buscond@microsoft.com) **寻求帮助。**

在事情不对劲时直言不讳，以便我们解决问题。

报告诚信问题



# 如何 报告问题

提出问题应尽可能简单。因此我们提供多种报告异常状况的方法。

选择一种你能够运用自如的报告方法。  
无论你选择哪种方法，我们都会为你保密。

如果你所在国家/地区的法律允许，你可以通过我们的外部热线匿名报告，你也可以要求合规和道德团队将你的报告作为匿名报告处理。



**网站** [www.microsoftintegrity.com](http://www.microsoftintegrity.com)



**传真** 1-425-708-7177



**电子邮件地址**

[buscond@microsoft.com](mailto:buscond@microsoft.com)



**邮寄地址**

Business & Regulatory Investigations  
Microsoft Corporation  
One Microsoft Way  
Redmond, WA 98052 United States



**电话** 1-877-320-6738

国际接听方付费电话  
+1-720-904-6844

## 其他资源

此外，你还可以随时向你的经理、任何 Microsoft 管理人员、人力资源、财务或 CELA 提出问题。

## 正在使用移动设备？

扫描此二维码，  
向我们发送电子邮件。



SCAN ME

# 在事情不对劲时 直言不讳

Microsoft  
runs on  
trust.

要打造信任文化，我们需要学会在事情不对劲时直言不讳，以便我们解决这个问题。

## 提出 诚信 问题



提出诚信问题时，你不必掌握所有细节，也不必确信一定出现了某些状况。

我们将努力严肃、公平地对待你的报告，如果据我们了解确实有问题需要解决，我们将及时采取行动。





## 我们禁止报复行为

在事情不对劲时直言不讳，这需要勇气。我们知道你可能会感到不自在或担心。因此我们制定了禁止报复行为的政策。我们禁止对以下行为发起报复：



拒绝做出违反信任准则、Microsoft 政策或法律的行为，即使你的拒绝会导致 Microsoft 遭受业务损失。



对可能发生的不当行为，善意地表达自己的顾虑。



配合调查。

我们的策略规定，因员工从事任何此类活动而对其进行打击报复的人员都将受到惩戒处分，甚至包括解雇。



# 管理人员的 职责



**作为领导，  
你负有为团队打造  
文化和工作环境的  
特殊责任。**

```
app.MapGet( "/trust/{id}" , async.  
    await db.TrustCode.FindAsync(id  
        ? Results.Ok(trust)  
        : Results.NotFound());  
  
app.MapPost( "/trust" , async.(Trusted  
{  
    if (!MinimalValidation.TryVali  
        return Results.Validation  
    db.TrustCode  
    await db.S  
    return.Res  
    /trust/  
});  
app.MapPut( "/tr  
TrustedDb db) =>
```



你做出决策和处理问题、不同意见甚至坏消息的方式，将为你的团队、客户和利益相关者奠定信任的基础。你的成功和团队的成功取决于你们共同建立的信任。

采取这些简单的步骤，  
在团队中打造信任和诚信文化：

## 沟通

定期向你的团队阐明道德和诚信的重要性，并明确表达你期望团队成员以合乎道德的方式完成工作。



## 示范

展现符合道德标准的决策制定和良好的判断力，以身作则。



## 问题

确保你的团队知道，必须通过正确的方式取得重要的成果。然后，提出问题并确信已通过正确的方式取得了成果。



## 聆听

作为管理人员，你需要指导团队并表现出对团队的关心，这一点至关重要。确保你的团队知道，即使他们有什么难言之隐，你也会认真倾听。





# 管理人员应如何 处理问题

如果有人带着问题来找你，你负有聆听并采取行动的特殊责任。  
正确地处理问题对于维护信任和保护 Microsoft 至关重要。

如果有人提出问题，请遵照以下步骤：



消除干扰并**仔细聆听**。向直言不讳的报告者表达**感谢**，请记住，他们做了一件非常困难，但对 Microsoft 来说非常重要的事情。



以**尊重的态度**回应并**认真对待**每一个问题，即使你不同意他们的看法。表明你正在努力解决问题。



采取措施为报告问题的人员**保密**，避免与团队中的其他人讨论你们之间的对话。

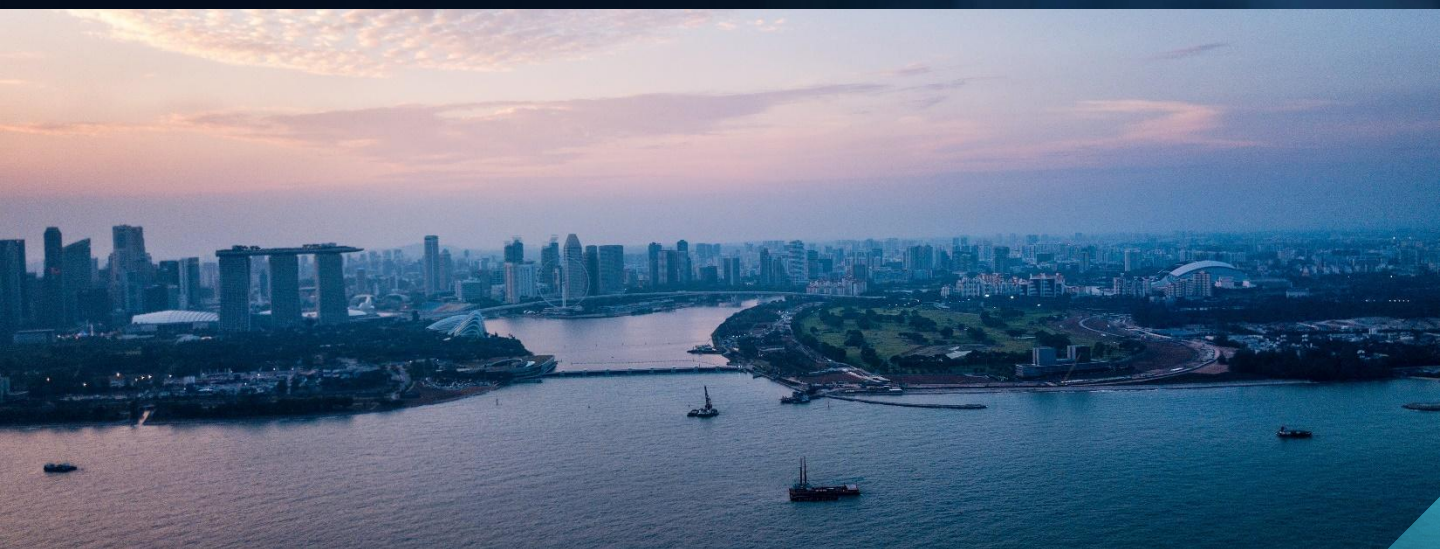


如果你得知一个问题或观察到一种行为可能违反**反骚扰**和**反歧视**政策，则必须及时联系 [AskHR](#)。

作为一名管理人员，你有能力自行解决与表现相关的问题，但你应该将与商业道德有关的诚信问题或不当行为上报给 Microsoft 商业和监管调查部门。

如果你有疑问或不确定自己是否能够或应该解决此问题，你可以随时向商业行为与合规部门（邮箱 buscond@microsoft.com）发送电子邮件以寻求建议。

[发送电子邮件至商业行为与合规部门 >](#)



在事情不对劲时直言不讳，以便我们解决问题。

[报告诚信问题](#)



[不当行为相关问题报告政策](#)



# TRUST

客户对我们的信任

尊重隐私

18

请勿进行不当付款

19

公平竞争

21

构建安全的产品和服务

22







在做决策时，问问自己，  
这会建立还是会损害客户  
对我们的信任？

能够随时随地满足人员和组织的需求，我们才能取得成功。

我们的一切行为都必须以我们的客户为核心。  
我们需要想方设法与客户互动、支持客户并帮助客户成就不凡。



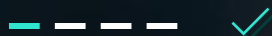


# 尊重 隐私

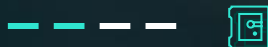
我们尊重隐私，以开诚布公的方式处理包括个人数据在内的客户数据。

我们希望客户信任我们会保护他们的隐私并以他们允许的方式使用其数据。

## <我们如何尊重隐私>



遵守当地隐私和数据保护法



在收集或处理个人数据（包括员工个人数据）时提供清晰准确的隐私通知。



尊重隐私选择，仅在提供客户同意的服务时使用客户数据。



通过构建安全的产品和服务来保护客户的数据。

访问以下链接，了解更多 ↓

[隐私政策（仅限内部）](#)



[隐私原则](#)







```
app.MapGet("/trust/incomplete", async (TrustedDb db) =>
await db.TrustCode.Where(t => !t.IsComplete).ToListAsync());
app.MapGet("/trust/{id}", async (int id, TrustedDb db) =>
await db.TrustCode.FindAsync(id) is Trusted trust
? Results.Ok(trust)
: Results.NotFound());
```

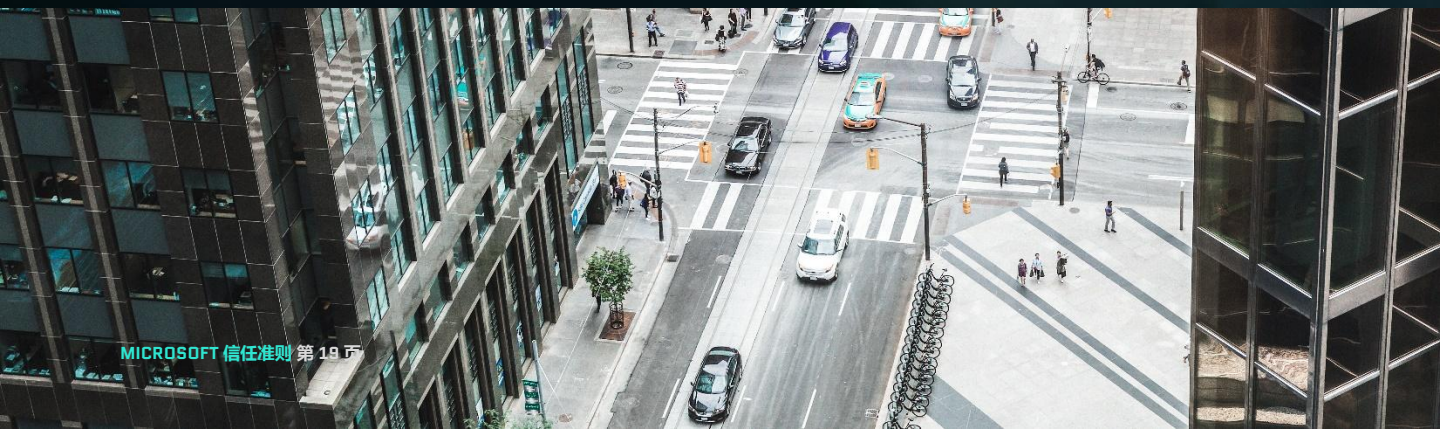
# 请勿进行 不当付款

我们承诺与客户进行真诚透明的互动，我们禁止任何通过不当付款来影响客户决策的尝试。

腐败会在很多方面损害我们的客户，包括有时导致客户付出超出他们应付金额的资金。

与通过不当开展业务而可能获得的任何收益相比，始终赢得并维护客户的信任更加重要。在世界上的某些地区，某些人可能会通过行贿来获取业务。但我们禁止这种行为。

我们宁愿失去业务，也不会通过贿赂、回扣或其他不正当利益来取得业务。





## <我们如何建立信任>

拒绝向任何人行贿或支付回扣。

禁止各种腐败付款，包括用于获得许可证或审批的付款，以及用于加快政府常规程序的小额付款（通常称为“疏通费”）。

在任何交易中，我们都将避免隐藏的条款或安排，并尽可能降低复杂性，因为透明交易可降低行贿或回扣的风险。

选择以诚信闻名的合作伙伴，并报告代表不符合道德规范或可能行贿的迹象。

确保我们提供给政府官员或客户的任何礼品、招待或旅行都是合理和适当的，并在必要时获得预先批准。

根据应聘者的优势招聘人员，而不是做出有利于客户或政府官员的招聘决策。

如果我们进行慈善捐赠，我们这样做的目的是支持合法的慈善事业，而不是作为利益交换的一部分。

反腐败政策



向政府官员提供礼品、招待和旅行  
(仅限内部)



反腐败和反贿赂承诺



```
async. (int id,TrustedDb db) =>  
e.FindAsync(id) is Trusted trust  
(trust)  
tFound());  
sync.(Trust  
) =>  
ation.Tr  
ults.Va  
rust);  
esAsyn  
ated(  
ed, TrustedDb db) =>  
olations))  
/a  
t  
1)
```

# 公平竞争

我们的竞争方式和我们取得的成果一样重要。

健康的竞争和公平的商业惯例使我们坚持客户至上，让客户能够以合理的价格获得各种产品和服务。公平竞争确保我们凭借诚信实现业务目标。

## 〈我们如何公平竞争〉

- 避免与竞争对手达成任何限制竞争的正式或非正式协议。
- 尊重竞争性投标流程，不操纵或安排结果，也不帮助其他人这样做。
- 不强行规定我们的独立渠道合作伙伴向客户收费的定价。
- 仅使用适当的资源收集竞争情报，如实报告且不虚报我们的身份。

访问以下链接，了解更多

公平竞争政策  
(仅限内部)



竞争情报政策  
(仅限内部)



# 构建安全的 产品和服务

我们致力于设计、构建、测试和运用我们的技术，以确保我们交付的解决方案符合最严格的安全标准。

我们努力保密客户数据的方方面面、维护数据完整性和安全性、确保我们的业务功能有效运行并能抵御攻击。我们所有人都担负着为 Microsoft 和客户的安全作出贡献的责任。

```
if (!MinimalValidation.TryValidate(trust, out var v
    return Results.ValidationProblem(violations);
db.TrustCode.Add(trust);
await db.SaveChangesAsync();

return Results.Created($"/trust/{trust.Id}", trust);
});

app.MapPut("/trust/{id}", async (int id, Trusted, input
{
    if (!MinimalValidation.TryValidate(inputTrusted, o
        return Results.ValidationProblem(violations);

    var trust = await db.TrustCode.FindAsync(id);

    if (trust is null) return Results.NotFound();

    app.MapGet("/trust/{id}", async (int id, TrustedDb
        await db.TrustCode.FindAsync(id) is Trusted trust
            ? Results.Ok(trust)
            : Results.NotFound());

    app.MapPost("/trust", async (Trusted trust, TrustedDb
    {
        if (!MinimalValidation.TryValidate(trust, out var v
            return Results.ValidationProblem(violations);
            db.TrustCode.Add(trust);
            await db.SaveChangesAsync();
            return Results.Created($"/trust/{trust.Id}", trust);
    });

    app.MapPut("/trust/{id}", async (int id, Trusted, input
    {
        if (!MinimalValidation.TryValidate(inputTrusted, o
            return Results.ValidationProblem(violations);

        var trust = await db.TrustCode.FindAsync(id);

        if (trust is null) return Results.NotFound();
    });
}
```





## <如何构建安全的产品和服务>

为安全核心优先级带来影响。

完成培训并应用所学的知识（包括安全基础和信任守则[商业行为标准]课程）。

了解 Microsoft 安全未来计划 (SFI) 的框架和目标。

在各自的工作中，优先考虑安全性，根据安全未来计划 (SFI) 提前规划安全措施并运用在产品、流程和工具中。

通过 [MSProtect](#) 详细了解如何参与保护 Microsoft。

通过立即报告功能，汇报任何可疑或恶意的网络活动、对我们系统和数据完整性的疑虑，以及任何可疑的数字或物理安全侵犯行为和漏洞。

### 访问以下链接了解详情



安全核心优先级（仅限内部）



Microsoft 安全政策（仅限内部）



安全未来计划



立即报告（仅限内部）



MSProtect（仅限内部）



# TRUST

## 政府和社区对我们的信任

尊重世界各地的法律

25

设计无障碍产品和服务

27

尊重并倡导人权

28

开发和部署负责任 AI

29

**在做决策时，请问问自己：这是否能促进政府和社区对我们的信任？**

要完成我们的使命，就需要让政府和社区知道，我们将遵守我们运营业务所在的国家/地区的法律和习俗，并尊重全球的人权。

```
app.MapGet( "/trust/{id}" ,    async.  
    await db.TrustCode.FindAsync(i  
    ? Results.Ok(trust)  
    : Results.NotFound());  
  
app.MapPost( "/trust" , async.(Truste  
{  
    if (!MinimalValidation.TryVali  
        return Results.Validation  
    db.TrustCode.Add(trust);  
    await db.SaveChangesAsync();  
    Results.Created($" /trust  
});  
app.M    /{id}" , async (int  
{  
    Validation, TryValida  
    Results.ValidationPro  
    = await db.TrustCod  
    is null) return Res
```







# 尊重

世界各地的

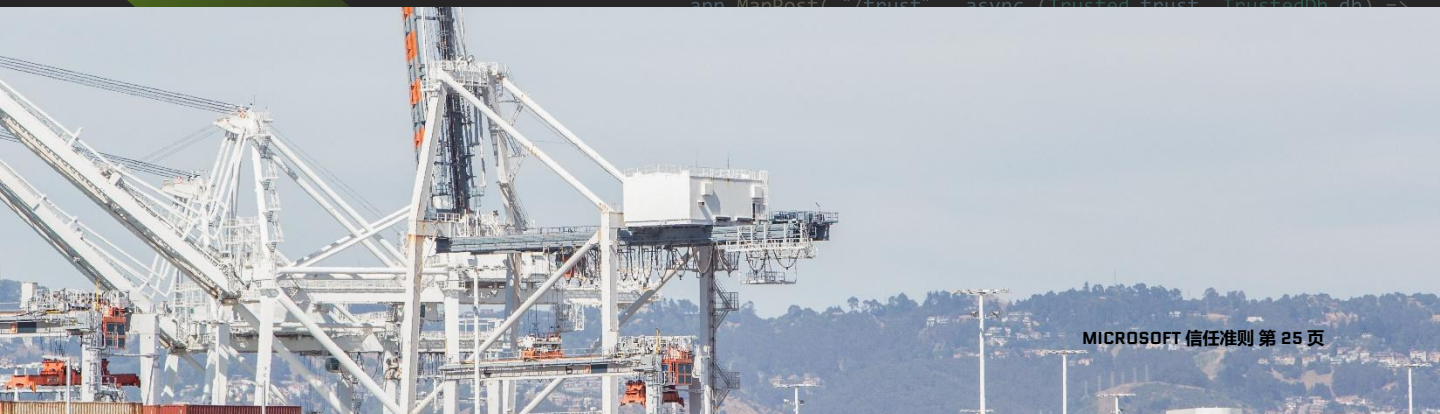
# 法律

我们的业务遍及全球，我们承诺遵守业务开展地的法律。

这有助于我们与政府建立信任，壮大我们生活和工作的社区，并为打造我们能在其中受到上述法律保护的公平社会提供助力。

```
app.MapGet("/trust/{id}", async (int id, TrustedDb db) =>
    await db.TrustCode.FindAsync(id) is Trusted trust
    ? Results.Ok(trust)
    : Results.NotFound());
```

```
app.MapPost("/trust", async (Trusted trust, TrustedDb db) =>
```







## <我们如何尊重世界各地的法律>



尊重和遵守禁止我们与某些国家、政府、实体和个人开展业务的全球贸易管制和经济制裁。



在与政府代表和官员的讨论中做到诚实透明。



如实、适当、及时地响应政府的信息查询请求。



遵守规定如何在国际上分销和使用我们的技术的相关法律。



遵守美国以及我们的所有业务运营地的法律和法规。



# 设计无障碍

## 产品和服务

我们在产品和服务开发中优先考虑包容性设计。

全世界有超过 10 亿残障人士，我们有机会并有责任进行包容性设计。对辅助功能的投资可帮助 Microsoft 支持客户实现包容性目标。我们认为，如果技术能够体现出运用技术的每个人的多元化，那么人们所能实现的成就无可限量。

### <我们如何设计无障碍产品>



从设计过程开始，就将不同观点的人纳入其中并向他们学习，进行包容性设计并以人为本。



让所有用户（包括存在视力、听力、学习、行动、心理健康、与年龄相关的、感官和语言障碍的用户）都能使用我们的产品。

Microsoft 辅助功能 >

辅助功能资源和培训 >

公共政策建议 >

# 尊重并倡导 人权

我们致力于通过尊重并倡导人权来使全世界每一个人都能够利用技术造福人类。

## <我们如何尊重并倡导人权>



我们尊重并提倡以国际法、原则和规范为依据的法治与人权。



我们认为，我们在面临重大人权挑战的国家/地区开展运营，并与这些国家/地区的人民、政府和其他利益相关者进行负责任的互动，从长远来看，往往会给改善人权带来更大的希望。



当我们面对政府提出的提供用户数据或删除内容的要求时，我们会核实政府的要求是否有效、具有法律约束力，并且遵守法律规定。



我们通过尽职调查来评估技术对人权的影响。我们向利益相关者学习，以反思我们的想法，制定和完善我们的政策和实践，改进我们的技术以及提供技术以履行人权承诺的方式。

我们的人权承诺





# 开发和部署 负责任 AI

我们致力于以符合 AI 原则、负责任 AI 标准、公司使命、价值观和人权承诺的方式开发和部署 AI 技术。

我们努力打造值得信赖的系统。这意味着将员工及其目标置于系统设计决策的中心，并尊重公平公正、安全可靠、保护隐私、开放包容、公开透明和恪守责任等持久价值观。

```
        return Results.ValidationProblem(violations);
        db.TrustCode.Add(trust);
        await db.SaveChangesAsync();
        return Results.Created($" /trust/{trust.Id}" , trust);
    });
    app.MapPut( "/trust/{id}" , async (int id, Trusted inputTru
    {
        if (!MinimalValidation.TryValidate(inputTrusted, out car
            return Results.ValidationProblem(violations);
        var trust = await db.TrustCode.FindAsync(id);
        if (trust is null) return Results.NotFound();

    app.MapGet( "/trust/{id}" , async (int id, TrustedDb db)
        await db.TrustCode.FindAsync(id) is Trusted trust
            ? Results.Ok(trust)
            : Results.NotFound());

    app.MapPost( "/trust" , async(Trusted trust, TrustedDb db)
    {
        if (!MinimalValidation.TryValidate(trust, out var violatio
            return Results.ValidationProblem(violations);
        db.TrustCode.Add(trust);
        await db.SaveChangesAsync();
        return Results.Created($" /trust/{trust.Id}" , trust);
    });
    app.MapPut( "/trust/{id}" , async (int id, Trusted inputTru
    {
        if (!MinimalValidation.TryValidate(inputTrusted, out car
            return Results.ValidationProblem(violations);
        var trust = await db.TrustCode.FindAsync(id);
        if (trust is null) return Results.NotFound();
```

## <如何开发和部署负责任 AI>

在开发和部署 AI 技术时遵循 **Microsoft 负责任 AI 标准**。

向负责任 AI 办公室报告敏感用途并遵循收到的指导。

在需要更多指导时寻求帮助。

访问以下链接，了解更多



**负责任 AI 办公室**  
(仅限内部)



**Microsoft 负责任 AI 政策**  
(仅限内部)



**报告敏感用途**  
(仅限内部)



**Microsoft 负责任 AI 原则**



**Microsoft 负责任 AI 标准**



# TRUST

## 我们对彼此的信任

促进多元与包容

32

安全高效的工作场所

34

避免利益冲突

36

**在做决策时，请问问自己：这能否改善我们基于 One Microsoft 文化的协作方式？**

成员彼此信任的团队，彼此开诚布公、互为助益的团队才能做出最大的贡献。

这需要我们所有人共同努力，营造相互尊重、包容和协作的氛围。

```
app.MapGet( "/trust/{id}" , async (int id) => {
    await db.TrustCode.FindAsync(id) is TrustCode
    ? Results.Ok(trust)
    : Results.NotFound());

app.MapPost( "/trust" , async (Trusted trust) => {
    if (!MinimalValidation.TryValidate(trust))
        return Results.ValidationProblem(validationProblems);
    db.TrustCode.Add(trust);
    await db.SaveChangesAsync();
    return Results.Created($" /trust/{trust.Id}" , trust);
});

app.MapPut( "/trust/{id}" , async (int id, TrustCode trust) => {
    if (!MinimalValidation.TryValidate(trust))
        return Results.ValidationProblem(validationProblems);
    var trust = await db.TrustCode.FindAsync(id);
    if (trust is null) return Results.NotFound();
    trust.Update(trust);
    await db.SaveChangesAsync();
    return Results.NoContent();
});

app.MapPut( "/trust/{id}" , async (int id, TrustCode trust) => {
    if (!MinimalValidation.TryValidate(trust))
        return Results.ValidationProblem(validationProblems);
    var trust = await db.TrustCode.FindAsync(id);
    if (trust is null) return Results.NotFound();
    trust.Update(trust);
    await db.SaveChangesAsync();
    return Results.NoContent();
});
```





# 促进 多元与包容

提出分歧而不是无视分歧，  
这样我们才能更好的合作。

我们相信，我们能够为全球  
每一个人提供出色的服务，  
并通过我们每个人带来的多元技能、  
经验和背景来丰富自己的文化。

```
return Results.ValidationProblem(V
db.TrustCode.Add(trust);
await db.SaveChangesAsync();
return Results.Created($" /trust/{trust.Id}");
});
app.MapPut( "/trust/{id}" , async (int id, Trust
{
    if (!MinimalValidation.TryValidate(inputTru
        return Results.ValidationProblem(violat
        var trust = await db.TrustCode.FindAsync
        if (trust is null) return Results.NotFound

app.MapGet( "/trust/{id}" , async (int id, T
    await db.TrustCode.FindAsync(id) is Trust
        ? Results.Ok(trust)
        : Results.NotFound());

app.MapPost( "/trust" , async (Trusted trust, Tr
{
    if (!MinimalValidation.TryValidate(trust
        return Results.ValidationProblem(V
        db.TrustCode.Add(trust);
        await db.SaveChangesAsync();
        return Results.Created($" /trust/{trust.Id}");
    });
    app.MapPut( "/trust/{id}" , async (int id, Trust
    {
        if (!MinimalValidation.TryValidate(inputTru
            return Results.ValidationProblem(violat
            var trust = await db.TrustCode.FindAsync
            if (trust is null) return Results.NotFound
```



## <我们如何促进多元与包容>



为所有符合条件的应聘者  
和雇员提供平等的就业机会。



始终承诺雇用残障人士，  
并为符合条件的员工提供  
合理的工作调适。



审视我们的无意识偏见，  
并采取措施来打造一种包  
容性文化，让所有员工都  
感觉自己受欢迎。



不因年龄、血统、肤色、家庭或医疗  
休假、性别认同或表达、遗传信息、  
婚姻状况、健康状况、民族出身、身  
心残疾、政治派别、受保护的退伍军  
人身份、种族、宗教、性别（包括怀  
孕）、性取向或受适用法律、法规和  
条例保护的任何其他特征而歧视他人。



访问以下链接，了解更多 ↓

[全球多元化和包容性](#)



[平等就业机会政策（仅限内部）](#)







# 为打造安全 高效的 工作场所 做出贡献

我们致力于创造一种能够让每个人出色工作的工作环境。安全的工作场所能够激发信任，让所有人都能做出贡献并取得成功。

存在歧视和骚扰且不安全的工作环境会降低我们本可以共同取得的成就。

访问以下链接，了解更多 ↓

反骚扰和歧视政策  
(仅限内部)



招待政策 (仅限内部)



工作场所暴力政策  
(仅限内部)



无毒品工作场所政策  
(仅限内部)



```
app.MapGet("/trust/{id}", async (int id, TrustedDb db) =>
{
    await db.TrustCode.FindAsync(id) is Trusted trust
    ? Results.Ok(trust)
    : Results.NotFound());

app.MapPost("/trust", async (Trusted trust, TrustedDb db) =>
{
    if (!MinimalValidation.TryValidate(trust, out var violations))
        return Results.ValidationProblem(violations);
    db.TrustCode.Add(trust);
    await db.SaveChangesAsync();
    return Results.Created($" /trust/{trust.Id}", trust);
});

app.MapPut("/trust/{id}", async (int id, Trusted inputTrusted, TrustedDb db) =>
{
    if (!MinimalValidation.TryValidate(inputTrusted, out var violations))
        return Results.ValidationProblem(violations);
    var trust = await db.TrustCode.FindAsync(id);
    if (trust is null) return Results.NotFound();
    trust = inputTrusted;
    db.TrustCode.Update(trust);
    await db.SaveChangesAsync();
    return Results.NoContent();
});
```



```
app.MapGet( "/trust/{id}", async (int id,TrustedDb db) =>
    await db.TrustCode.FindAsync(id) is Trusted trust
    ? Results.Ok(trust)
    : Results.NotFound());

app.MapPost( "/trust", async.(Trusted trust, TrustedDb db) =>
{
    if (!MinimalValidation.TryValidate(trust, out var violations))
        return Results.ValidationProblem(violations);
    db.TrustCode.Add(trust);
    await db.SaveChangesAsync();
    return Results.Created($" /trust/{trust.Id}" , trust);
});
app.MapPut( "/trust/{id}", async (int id, Trusted. inputTrusted, TrustedDb db) =>
{
    if (!MinimalValidation.TryValidate(inputTrusted, out var violations))
        return Results.ValidationProblem(violations);
    var trust = await db.TrustCode.FindAsync(id);
    if (trust is null) return Results.NotFound();
```

## <如何为打造安全高效的工作场所做出贡献>

— — — —

将良好的安全与健康实践整合到我们的运营中并遵守工作场所安全法规。

— — — —

尊重他人，绝不容忍骚扰或歧视。

— — — —

在尊重他人的前提下解决问题，绝不诉诸暴力行为或威胁。

— — — —

不在摄入毒品或酒精的情况下工作。

# 避免 利益 冲突

当我们的个人关系或财务利益与我们的工作职责重叠时，可能会出现利益冲突。

如果我们不能谨慎处理潜在的利益冲突，这些情形会影响我们做出的决定，削弱团队内部的信任，并损害 Microsoft 的声誉。

```
app.MapGet( "/trust/{id}" , async (int id,TrustedDb db) =>
{
    await db.TrustCode.FindAsync(id) is Trusted trust
    ? Results.Ok(trust)
    : Results.NotFound());

app.MapPost( "/trust" , async (Trusted trust, TrustedDb db) =>
{
    if (!ModelState.IsValid)
    {
        return Results.BadRequest(ModelState);
    }
    db.TrustCode.Add(trust);
    await db.SaveChangesAsync();
    return Results.CreatedAt("/trust", db.TrustCode, trust);
});

app.MapPut( "/trust/{id}" , async (int id,Trusted trust, TrustedDb db) =>
{
    if (!ModelState.IsValid)
    {
        return Results.BadRequest(ModelState);
    }
    var trust = await db.TrustCode.FindAsync(id);
    if (trust == null)
    {
        return Results.NotFound();
    }
    trust.TrustCode = trust.TrustCode.Replace(trust.TrustCode, trust.TrustCode);
    db.TrustCode.Update(trust);
    await db.SaveChangesAsync();
    return Results.NoContent();
});
```



## <我们如何避免利益冲突>



按照 Microsoft 的最佳利益行事。



如果在某些情形下，我们在另一家公司中存在的个人关系或财务利益可能会影响我们在工作中做出的决策，则应避免或规避这些情形。

(列表在下一页继续) ↓



了解即使我们相信我们的决策不会受到外部关系的影响，利益冲突仍然存在。



当出现利益冲突时，我们会披露这些冲突，并获得建议或批准。



在开展外部工作之前，我们会确保工作不会损害 Microsoft 的商业利益，也不会破坏我们签署的任何员工协议。

访问以下链接，了解更多



```
app.MapGet( "/trust/{id}" , async (int id,TrustedDb db) => {
    await db.TrustCode.FindAsync(id) is Trusted trust
    ? Results.Ok(trust)
    : Results.NotFound();
})
```

**利益冲突**  
(仅限内部)



**外部工作政策**  
(仅限内部)



**利益冲突审批工具**  
(仅限内部)





# TRUST

## 投资者和公众对我们的信任

勿将内部信息作为交易筹码

39

保存准确的记录和合同

41

向公众传达准确的信息

43

保护 Microsoft 的资源

45

保护机密信息和知识产权

46

**在做决策时，请问问自己：**

**这会提升还是会损害 Microsoft 在投资者和公众中的声誉？**

我们的投资者务必要相信我们会以负责任且周到的方式为他们创造价值。

通过保护我们的资产和信息，准确说明我们公司的目标，并与他们和公众诚实地沟通，我们将能够赢得他们的信任。



```
db.TrustCode.Add(trust);
await db.SaveChangesAsync();
return Results.Created($" /trust/{id}", trust);
});
app.MapPut( "/trust/{id}" , async (id, trust) =>
{
    if (!MinimalValidation, TryValidateModel(trust))
        return Results.ValidationProblem();
    var trust = await db.TrustCodes.FindAsync(id);
    if (trust is null) return Results.NotFound();
    trust.TrustCode = trustCode;
    await db.SaveChangesAsync();
    return Results.Created($" /trust/{id}", trust);
});
}
```

# 勿将内部信息 作为交易筹码

为了维持投资者和公众的信任，我们遵守证券法，这意味着我们不会在掌握重大非公开信息的情况下进行交易。

在工作中，我们可能会了解到有关 Microsoft 或有业务往来的公司的重大非公开信息。如果一位理性的投资者认为在决定买卖公司证券时某些信息很重要，那么这些信息就是“重大信息”。如果信息尚未广泛传达给投资公众，则信息是“非公开的”。

根据此信息交易 Microsoft 股票或任何其他公司的股票，不仅会破坏投资者和公众的信任，而且是违法行为，违反了公司政策，并且对其他投资者也不公平。



访问以下链接，了解更多 ↓

内幕交易政策  
(仅限内部)



限制交易窗口政策  
(仅限内部)



# <我们如何建立信任>

```
app.MapGet( "/trust/{id}" , async (id, trustedDb db) =>
    await db.TrustCode.FindAsync(id)
    ? Results.Ok(trust)
    : Results.NotFound() );

app.MapPost( "/trust" , async (trust, trustedDb db) =>
{
    if (!MinimalValidation(trust, out var violations))
        return Results.BadRequest(em(violations));
    db.TrustCode.Add(trust);
    await db.SaveChangesAsync();
    return Results.Created($"trust/{trust.Id}", trust);
});
```



绝不依据 Microsoft 的重大非公开信息或通过我们的雇佣关系获得的重大非公开信息购买或出售任何证券。



绝不向他人（例如朋友、配偶或经纪人）提供有关重大非公开信息的“提示”。



绝不建议其他人依据 Microsoft 的重大非公开信息或通过我们的雇佣关系获得的重大非公开信息进行证券交易，即使我们不共享该信息也是如此。





# 保存准确的 记录和交易

我们的记录清晰、准确、完整。

投资者依靠准确且易于理解的信息来了解我们的财务业绩以及我们公司的发展方向，并且对这一方向充满信心。

保存准确的记录对于维护投资者的信任、做出正确的商业决策和履行我们作为上市公司的监管义务至关重要。



## <我们如何保存准确的记录和交易>



我们按照适用的法律、法规和会计惯例诚实、准确地记录和报告财务交易和业务信息



我们对合作伙伴和客户做出公开承诺，我们的书面合同反映了交易的实际经济意义。



我们不会签订附属协议或做出其他“账外”安排



在保留和处置文档时，我们遵循既定的公司保留要求，从而履行我们的法律义务并满足长期业务需求。



我们仅在获得适当授权的情况下执行财务交易，并按照 Microsoft 的会计惯例进行记录。



财务组织的首席执行官 (CEO)、首席财务官 (CFO)、公司控制官和其他员工遵守职业行为财务准则。



我们使用标准协议和合同流程或获取自定义条款审批。

访问以下链接，了解更多



职业行为财务准则  
(仅限内部)



自定义和附属协议政策  
(仅限内部)



文档保留政策  
(仅限内部)





## 向公众 传达准确的信息

我们的公开声明和通信诚实可信，从不误导他人。

我们所说的话以及我们的说话方式非常重要。只有投资者和公众知道自己可以依靠我们所做的承诺，我们才能赢得并维护投资者和公众的信任。





## <如何向公众传达准确的信息>



我们在提交给美国证券交易委员会的报告和文件以及其他公开声明中全面、准确、及时地进行财务披露。



我们认真倾听投资者和其他利益相关者的反馈和问题。



除非得到公司授权，否则我们不会代表公司发言；我们应将分析师和媒体询问交给投资者关系或公共关系部门来处理。



如果我们在社交媒体上宣传产品和服务，我们应明确公司隶属关系，并表明我们对宣传的产品或服务的真实看法或观点。



我们确保围绕产品和服务传达的任何信息都真实准确。



我们的广告和宣传材料准确无误，没有虚假表述。

访问以下链接，了解更多 ↓

**财务信息披露政策**  
(仅限内部)



**公共宣传政策**  
(仅限内部)



**社交媒体指导原则**  
(仅限内部)





# 保护

## Microsoft 的资源

我们对 Microsoft 的资源负有管理责任。我们每天都利用公司资源和资金来完成工作。这些资源属于 Microsoft；以维护信任且增加价值的方式负责任地使用这些资源，是我们对投资者的责任。

### <我们如何保护 Microsoft 的资源>

了解与业务相关的数据是 Microsoft 的财产，必须按照我们的政策和适用的法律来访问、保存和审查这些数据。

精打细算地花费 Microsoft 的资金，杜绝浪费和滥用行为。

将我们的公司帐户和服务（而不是个人帐户或服务）用于商业通信和数据。

在按照公司政策使用公司提供的技术资源并且是出于个人原因使用这些技术资源时做出正确的判断，仅在不会干扰我们的工作职责或损害我们的工作环境的前提下进行这种使用。

访问以下链接，了解更多 ↓

负责任的技术运用政策  
(仅限内部)



开支政策  
(仅限内部)



# 保护机密信息 和知识产权



**我们保护和尊重信息和创意的商业价值，**  
无论它们是属于 Microsoft 还是另一家公司。

**如果我们披露 Microsoft 的机密信息，可能会失去竞争优势。**

如果我们披露从有业务往来的公司获取的机密信息，就会失去信任。保护信息和创意（无论是我们自己的还是其他公司的）对于我们的业务成功至关重要，且有助于我们建立值得信赖的合作伙伴的声誉。



## <我们如何保护机密信息和知识产权>



使用最佳实践来保护对机密信息的访问，避免在公共空间或与不需要了解机密信息的同事讨论机密信息。



即使我们从 Microsoft 离职，也不能将机密信息用于非 Microsoft 商业用途并保持机密性。



通过版权、商标以及可保护创新价值的其他形式的知识产权来保护 Microsoft 自己的知识产权。



尊重他人的保密性和知识产权，未经授权不得擅自使用他人机密信息。



未经 CELA 许可或允许，不得使用或复制受版权保护的第三方材料，如软件、图形、视频和音乐等。

访问以下链接，了解更多 ↓

**机密信息**  
政策（仅限内部）



**版权政策**  
（仅限内部）



# TRUST

## 我们的代表对我们的信任

启用值得信赖的代表

49

负责任地处理礼品、招待和旅行

51

诚信地选择供应商

53

**在做决策时，请问问自己：  
这能否增强我们与代表和客户的长期工作关系？**

我们以高道德标准要求自己，我们希望我们的代表（包括合作伙伴、供应商、顾问和代表我们的其他公司）能够达到我们为自己设定的标准。

当我们将价值回馈给代表，并且不要求他们做我们自己也不愿做的事情时，就会赢得他们的信任。





## 启用 值得信赖的 代表

我们力争启用诚实且合乎道德标准的代表，他们承诺在代表我们工作时以合乎道德标准的方式开展业务。我们的业务依靠庞大的代表（如合作伙伴、供应商和顾问）网络来取得成功。

他们的行为在反映我们公司的声誉方面发挥的作用与我们自身的行为并无二致。当我们的代表诚信地开展业务时，我们就能维护并增强与客户和公众建立的信任。





## <我们如何启用值得信赖的代表>



在聘请他们代表我们工作之前，了解代表的资格和声誉。



了解让特定代表参与交易的原因，并且只有在适当的情况下才这样做。



培养代表对我们的信任，从不要求或建议他们做法律或我们的政策不允许的事情。



不向合作伙伴或经销商施压，要求他们订购不想要或不需要的产品或服务，如果他们拒绝订购，不对他们进行报复。



报告代表在代表 Microsoft 工作时可能从事腐败或不道德行为的迹象。

访问以下链接，了解更多 ↓

“值得信赖的代表”政策  
(仅限内部)



“以合乎道德的方式销售”政策  
(仅限内部)



# 负责任地 处理礼品、招待和旅行

我们努力在互致商务礼待时做出正确的判断并温和友善。

赠送和接受礼品、招待和旅行可以在 Microsoft 和有业务往来的公司之间建立牢固的工作关系和商誉。

但是，奢侈或缺乏透明度或合法目的的礼品、餐食或旅行可能被视为贿赂或完全不恰当。这些会损害信任并损害我们的业务。



## <我们如何负责任地处理礼品、招待和旅行>



发展不存在腐败和商业贿赂的积极关系。



在提供或接受礼品、招待或旅行时，确保它们合理、适当且具有合法的商业目的。



不向第三方索要礼品、招待或旅行，也不得让他们感到有义务提供一些福利才能与我们开展业务。



不得要求代表（如合作伙伴或供应商）来代表我们赠送礼品、招待或旅行。



不提供也不接受现金。

访问以下链接，了解更多



向业务对象提供礼品、招待和旅行的政策（仅限内部）



接受礼品、招待和旅行的政策（仅限内部）



向政府官员提供礼品、招待和旅行的政策（仅限内部）





# 诚信地选择 供应商

我们依靠供应商关系取得成功。为了完成使命，我们需要像我们一样竭尽全力与客户建立信任的供应商，他们应出色地完成工作并且遵守法律。



## <如何诚信地选择供应商>

选择适合这份工作的最佳供应商，并仔细考虑其可靠的业绩记录、诚信声誉和其他优点，而不是基于喜好而选择。

努力消除供应链中的童工、人口贩卖和其他劳动力滥用行为。

访问以下链接，了解更多



**供应商承包政策**  
(仅限内部)



**采购政策**  
(仅限内部)



# 坚持我们的 TRUST CODE



## 员工的责任

员工应遵守此信任准则、Microsoft 政策和法律。这意味着你必须：

**阅读、理解和遵守本信任准则以及适用于你的工作的政策、法律和法规。**

**在看到可能违反信任准则、Microsoft 政策以及法律和监管要求的行为时直言不讳。**

**在任何内部调查中诚实充分地配合。不隐瞒或销毁信息。**

**完成关于信任准则的培训，并证明你理解并承诺遵守信任准则。未能阅读或证明遵守信任准则，并不能让你免除履行这些责任。**

Microsoft 的商业行为准则及其合规性和道德计划得到了 Microsoft 董事会的认可和全力支持。董事会和管理层负责监督合规性和道德计划以及这些标准的遵守情况。这些标准适用于 Microsoft、其董事会以及 Microsoft 的所有员工、董事、执行官。

当我们提到 “Microsoft” 时，我们指的是 Microsoft Corporation 及其直接或间接拥有 50% 以上的投票控制权的所有子公司和附属公司。提到 “员工” 时，则包括 Microsoft 的董事会、员工、董事和高管。

只有董事会才能免除董事或执行董事遵守这些规定的规定。若要豁免董事或执行董事执行这些标准中的规定，将按照纳斯达克上市要求和适用的法律、规则和法规的要求公开披露。

这些标准对我们很重要。违反这些标准可能导致惩戒处分，甚至包括终止雇佣。员工应始终运用常识和良好的判断力，并且无论是否制定了禁止特定行为的政策，都应负责地行事。

在事情不对劲时直言不讳，以便我们解决问题。

**报告诚信问题**



# TRUST CODE

MICROSOFT 商业行为准则