

Links & UTM parameters

The three kinds of link in a Baziness integration, and how to tag the links that bring players to your game so every player is credited to the right campaign.

Three kinds of link

Link	Direction	Built by	Details
Baziness page links (sign-in, profile, <code>/shop</code> , <code>/game-offer</code>)	game → Baziness	your client, with settings from your server	Baziness pages
Payment page link (<code>/buy</code>)	game → Baziness Pay	your server only	Baziness Pay
Links into your game (<code>?utm_...</code> , <code>?tag=</code> , <code>?linkedToken=</code>)	outside → game	marketing, ads, the Baziness app	this page

Links into your game

Every link that brings players to your game should say **where they came from**:

Text

```
https://yourgame.example.com/?  
utm_source=baziness&utm_medium=inapp&utm_campaign=shahrivar_1405&utm_content=banner_home  
https://yourgame.example.com/?tag=telegram
```

Parameter	Meaning	Examples
<code>utm_source</code>	who sends the traffic	<code>baziness</code> , <code>telegram</code> , <code>instagram</code>
<code>utm_medium</code>	the kind of channel	<code>inapp</code> , <code>banner</code> , <code>cpc</code> , <code>social</code> , <code>push</code>
<code>utm_campaign</code>	the campaign	<code>shahrivar_1405</code> , <code>nowruz_1405</code>
<code>utm_content</code>	the creative or placement	<code>banner_home</code> , <code>story_2</code>
<code>tag</code>	a short channel tag (simpler alternative)	<code>telegram</code> , <code>google</code>
<code>linkedToken</code>	added by the Baziness app when it opens your game	the player's user token (Login)

Important

UTM values must be **plain ASCII**: lowercase letters, digits, `.` , `_` , `-` , at most 64 characters. A Persian campaign name is lost, so write `shahrivar_1405` , not `شهریور` .

How your game should read them

1. **Read them first**, before anything renders. Match the names case-insensitively.
2. **Remove them from the address bar**. The player doesn't see or edit them, and a player sharing their address doesn't credit your campaign with their friends.
3. **Store them on the device the first time** ("first touch"), and only if nothing is stored yet. A Baziness sign-in or an account switch restarts the game, and without local storage the tags would be lost.
4. **Send them with the first request that creates the player**, both your normal sign-in and `/api/platform/token` , whichever runs first.
5. **On the server, save them only when the player is created**. Never overwrite them later.
6. **Keep four separate fields** (`utm_source` , `utm_medium` , `utm_campaign` , `utm_content`), not one joined string. Campaign names contain underscores, so a joined value can't be split back.
7. **Leave `linkedToken` alone**. It isn't a campaign tag: the sign-in logic uses it, and removes it on the next restart.

Making campaign links

Give marketing a fixed recipe, or a small link generator, so every link carries **all four** parameters spelled the same way: your game's address, plus `utm_source`, `utm_medium`, `utm_campaign` and (optionally) `utm_content`, all in lowercase ASCII.

Test your links

- Open a campaign link in a private window. The address bar must be clean after the page loads.
- Sign in with Baziness, or trigger an account switch. The new player must still carry the campaign.
- Open the same link with an **existing** player. Their stored campaign must not change.

Ready-made code: [reading the tags and saving them](#), [building links](#).